Quantifying and Interpreting Uncertainty in Time Series Forecasting

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Kaleb Phipps

Tag der mündlichen Prüfung:	12. Dezember 2023
Referenten:	Prof. Dr. Veit Hagenmeyer Karlsruher Institut für Technologie (KIT)
	Prof. Dr. Alexander Mitsos RWTH Aachen University
	Prof. Dr. Ralf Mikut Karlsruher Institut für Technologie (KIT)

Kaleb Phipps
Quantifying and Interpreting Uncertainty in Time Series Forecasting
Doctoral Thesis
Revisions: June 3, 2024
Date of Examination: December 12, 2023
Reviewers: Prof. Dr. Veit Hagenmeyer, Prof. Dr. Alexander Mitsos, and Prof. Dr. Ralf Mikut

Karlsruhe Institute of Technology (KIT)
Institute for Automation and Applied Informatics (IAI)
Department of Computer Science
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen

List of Publications

Main Publications

K. Phipps, S. Lerch, M. Andersson, R. Mikut, V. Hagenmeyer, and N. Ludwig, "Evaluating ensemble post-processing for wind power forecasts", *Wind Energy*, vol. 25, no. 8, pp. 1379–1405, 2022. DOI: 10.1002/we.2736

K. Phipps, N. Ludwig, V. Hagenmeyer, and R. Mikut, "Potential of ensemble copula coupling for wind power forecasting", in *Proceedings 30. Workshop Computational Intelligence*, vol. 26, KIT Scientific Publishing, 2020, p. 87. DOI: 10.5445/IR/1000127955

K. Phipps, B. Heidrich, M. Turowski, M. Wittig, R. Mikut, and V. Hagenmeyer, "Generating probabilistic forecasts from arbitrary point forecasts using a conditional invertible neural network", *Applied Intelligence*, 2024. DOI: https://doi.org/10.1007/s10489-024-05346-9

K. Phipps, S. Meisenbacher, B. Heidrich, M. Turowski, R. Mikut, and V. Hagenmeyer, "Loss-customised probabilistic energy time series forecasts using automated hyperparameter optimisation", in *Proceedings of the Fourteenth ACM International Conference on Future Energy Systems*, ACM, 2023, pp. 271–286. DOI: 10.1145/3575813.3595204

K. Phipps, K. Schwenk, B. Briegel, R. Mikut, and V. Hagenmeyer, "Customized uncertainty quantification of parking duration predictions for EV smart charging", *IEEE Internet of Things Journal*, pp. 1–1, 2023. DOI: 10.1109/JIOT.2023.3299201

Further Publications

K. Schwenk, K. Phipps, B. Briegel, V. Hagenmeyer, and R. Mikut, "A benchmark for parking duration prediction of electric vehicles for smart charging applications", in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2021, pp. 1–8. DOI: 10.1109/SSCI50451.2021.9660063

B. Heidrich, A. Bartschat, M. Turowski, O. Neumann, K. Phipps, S. Meisenbacher, K. Schmieder, N. Ludwig, R. Mikut, and V. Hagenmeyer, "pyWATTS: Python workflow automation tool for time series", 2021. arXiv: 2106.10157

M. Turowski, B. Heidrich, K. Phipps, K. Schmieder, O. Neumann, R. Mikut, and V. Hagenmeyer, "Enhancing anomaly detection methods for energy time series using latent space data representations", in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ACM, 2022, pp. 208–227. DOI: 10.1145/3538637.3538851

S. Meisenbacher, M. Turowski, K. Phipps, M. Rätz, D. Müller, V. Hagenmeyer, and R. Mikut, "Review of automated time series forecasting pipelines", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 6, e1475, 2022. DOI: 10.48550/arXiv.2202.01712

B. Heidrich, M. Turowski, K. Phipps, K. Schmieder, W. Süss, R. Mikut, and V. Hagenmeyer, "Controlling non-stationarity and periodicities in time series generation using conditional invertible neural networks", *Applied Intelligence*, pp. 1–18, 2022. DOI: 10.1007/s10489-022-03742-7

M. Turowski, M. Weber, O. Neumann, B. Heidrich, K. Phipps, H. K. Çakmak, R. Mikut, and V. Hagenmeyer, "Modeling and generating synthetic anomalies for energy and power time series", in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ACM, 2022, pp. 471–484. DOI: 10.1145/3538637.3539760

B. Heidrich, L. Mannsperger, M. Turowski, K. Phipps, B. Schäfer, R. Mikut, and V. Hagenmeyer, "Boost short-term load forecasts with synthetic data from transferred latent space information", in *Proceedings of the 11th DACH+ Conference on Energy Informatics*, vol. 5, SpringerOpen, 2022, pp. 1–20. DOI: 10.1186/s42162-022-00214-7

M. Beichter, K. Phipps, M. M. Frysztacki, R. Mikut, V. Hagenmeyer, and N. Ludwig, "Net load forecasting using different aggregation levels", in *Proceedings of the 11th DACH+ Conference on Energy Informatics*, vol. 5, SpringerOpen, 2022, pp. 1–21. DOI: 10.1186/s42162-022-00213-8

M. Luh, K. Phipps, A. Britto, M. Wolf, M. Lutz, and J. Kraft, "High-resolution real-world electricity data from three microgrids in the global south", in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ACM, 2022, pp. 496–514. DOI: 10.1145/3538637.3539763

T. Gneiting, D. Wolffram, J. Resin, K. Kraus, J. Bracher, T. Dimitriadis, V. Hagenmeyer, A. I. Jordan, S. Lerch, K. Phipps, *et al.*, "Model diagnostics and forecast evaluation for quantiles", *Annual Review of Statistics and Its Application*, vol. 10, 2023. DOI: 10.1146/annurev-statistics-032921-020240

B. Heidrich, K. Phipps, O. Neumann, M. Turowski, R. Mikut, and V. Hagenmeyer, "ProbPNN: Enhancing deep probabilistic forecasting with statistical information", 2023. arXiv: 2302.02597

D. Werling, M. Beichter, B. Heidrich, K. Phipps, R. Mikut, and V. Hagenmeyer, "The impact of forecast characteristics on the forecast value for the dispatchable feeder", in *Companion Proceedings of the 14th ACM International Conference on Future Energy Systems*, ACM, 2023, pp. 59–71. DOI: 10.1145/3599733.3600251

R. Poppenborg, K. Phipps, H. Khalloof, K. Förderer, R. Mikut, and V. Hagenmeyer, "Dynamic chromosome interpretation in evolutionary algorithms for distributed energy resources scheduling", in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, ACM, 2023, pp. 755–758. DOI: 10.1145/3583133.3590666

D. Werling, M. Beichter, B. Heidrich, K. Phipps, R. Mikut, and V. Hagenmeyer, "Automating value-oriented forecast model selection by meta-learning: Application on a dispatchable feeder", in *Proceedings of the Energy Informatics.Academy Conference 2022 (EI.A 2022)*, in press, SpringerOpen, 2023

Abstract

For a wide variety of sectors, including energy, retail, and mobility, time series data is increasingly gaining importance. Within these sectors, critical applications include dispatch management in energy systems, warehouse storage optimisation in the retail sector, and traffic congestion management within the mobility sector. However, for such applications to be successful, they require reliable and trustworthy forecasts of the relevant time series. Unfortunately, any forecast of the future contains an inherent component of uncertainty. Therefore, to ensure these forecasts are trustworthy, they should quantify this uncertainty, i.e., probabilistic forecasts. However, quantifying this uncertainty through a probabilistic forecast may not sufficiently increase the trust in the forecast. The quantified uncertainty should also be interpreted in a manner that is useful for the considered application.

Therefore, the present dissertation takes a holistic approach by considering both quantifying and interpreting uncertainty in time series forecasts. To quantify uncertainty, we first investigate whether the meteorological uncertainty affecting many time series can be linked to the uncertainty in the forecast time series. We show that this link can be established, but post-processing is required to generate calibrated probabilistic forecasts. Second, we consider if the unknown data distribution of a time series can be used to include uncertainty in a forecast. Thereby, we present a novel approach for generating probabilistic forecasts from arbitrary point forecasts using a conditional invertible neural network and show how our approach outperforms benchmark probabilistic forecasts on common evaluation metrics. Third, we extend this approach using automated hyperparameter optimisation to generate probabilistic forecasts whose properties can be customised depending on the loss metric considered. This customisation occurs without retraining the underlying forecasting model and can further increase trust in the forecast by providing probabilistic forecasts tailored to specific requirements.

To interpret the uncertainty, we first introduce an approach that explains the origins of uncertainty in a probabilistic forecast using existing methods from explainable artificial intelligence. Our method is applicable to a wide range of probabilistic forecasting models, and we show that the resulting explanations deliver valuable insights. Second, we investigate regions of uncertainty that are particularly critical for mobility applications. We further propose various representations of this quantified uncertainty, which highlight these critical regions and can be particularly useful to the considered mobility application.

Overall, by considering multiple approaches to quantify and interpret uncertainty, this dissertation introduces multiple contributions that can be applied to increase trust in time series forecasts.

Acknowledgements

Over the last four years, I often doubted I would ever make it to this stage – to finally write this page...but now that I am here, all I can say is thank you! This journey would never have been possible without the support of a thousand pairs of hands, hands that helped me over rocky roads, lifted me up when I had fallen down, and pushed me forward when I couldn't go on.

First, I would like to thank Klemens Böhm, who organised my funding for the first three years of my PhD and provided constructive criticism through multiple retreats and one-on-one talks. My PhD would not have been possible without the support of my "doctor father" Veit Hagenmeyer – thank you for the advice, philosophical discussions, and chance to discover my passion for teaching during my time at the IAI. This dissertation would also never have been finished without the tireless support of my unofficial "doctor father" Ralf Mikut – thank you for the detailed feedback, the hours spent reading every page of every paper, and the enthusiastic discussions in regular meetings. I would also like to thank Alexander Mitsos for serving as a reviewer for this dissertation, Jürgen Beyerer for his role as an examiner in my doctoral exam, and Mehdi Tahoori and Kathrin Gerling for completing my examination committee.

Of course, this journey would not have been possible without funding. Therefore, I would like to thank both the German Research Foundation (DFG) for funding as part of the Research Training Group 2153 "Energy Status Data – Informatics Methods for its Collection, Analysis and Exploitation" and the Helmholtz Association Initiative and Networking Fund for funding via Helmholtz AI.

I also don't know how I would have made it through this journey without the assistance of the "problem solvers" at IAI - Andreas and Bernadette. Andreas – thank you for always solving difficult organisational problems and thank you Bernadette for always having an answer, no matter how difficult the question or how complicated the solution.

Throughout my PhD I was lucky to be surrounded by amazing colleagues. Thank you, therefore, to everyone I worked with both at IAI and other institutes, especially Andy, Anthony, Angelo, André, Benedikt, Claudia, Christian, Daniel, Dorina, Fabian, Friedrich, Frederik, Hossein, Ines, James, Jan, Johannes, Kai, Karl, Katharina, Klaus-Martin, Lisa, Lorenz, Luca, Marcel, Markus, Marian, Martha, Matthias, Max, Moritz, Nathalie, Nicole, Nils, Oli, Rafael, Rebecca, Roman, Sebastian, Simon, Stefan, Stephan, Tim, Tom, Vojtech, and Yanke. I would like to extend a special thanks to those colleagues who shared a room with me over the years. To Katharina, Marian, and Nicole – thank you for making the "Energiezimmer" such a welcoming place, being there during COVID, and filling my office hours with laughs, discussions, and creative drawings. To those who shared a room later in my journey - Max and Stefan - thank you for keeping the culture alive and

bringing ducks into my life. Also, to everyone who worked on the third floor – thank you for the good mornings, the coffee breaks, and the stories!

Throughout my PhD years, I had the chance to publish with many colleagues. To everybody who published with me, especially Dorina, Karl, Max, Rafael, Stefan, and Sebastian Lerch – thank you for the good times, great ideas, and successful publications. I would like to especially thank Benedikt, Kai, and Marian, who were there for most of the ideas, the highest of highs, and the lowest of lows - without your brains, company during all-nighters, and continuous support, I would never have finished this journey.

Special thanks goes to those colleagues who helped outside the academic realm. For the chocolate, coffee, and walks around campus north – thank you Claudia. For the questions, the Wednesday walks, and the crisis control – thank you Marian. Finally, for the laughs, the summer grill parties, and the incredibly long brunches – thank you to the "Ostdoktoranden".

Throughout this journey, I have been an incredible bundle of stress and nerves, and I am truly grateful for all the people who stood by me during the years. To my flatmates, Dennis, Ferdi, Henny, Lea, Simon, and Tobi – thank you for the early morning workouts during COVID, the home-cooked meals and the Fettschmelze pizza nights. For changing my training and changing my life – thank you Chris. To all my other friends who remain unnamed – thank you for the pick-me-ups, for giving me hell, for laughing with me, and for reminding me what is truly important.

Finally, I would like to thank my parents for believing in me even when I didn't believe in myself and Lisa for being there through the worst times with unwavering support and love that let me smile even when there was no light in sight.

To everyone – thank you. This dissertation is for all of you, I will never forget your support!

Kaleb Phipps, June 2024

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
$\mathrm{nMPI}(\beta)$	normalised Mean β-PI Width
BRR	Bayesian Ridge Regression
CDF	Cumulative Distribution Function
cINN	Conditional Invertible Neural Network
Conformal PI	Conformal Prediction Interval
CR	Coverage Rate
CRE	Coverage Rate Error
CRPS	Continuous Ranked Probability Score
CRPSS	CRPS Skill Score
CV	Cross Validation
ECMWF	European Centre for Medium-Range Weather Forecasts
EMOS	Ensemble Model Output Statistics
Empirical PI	Empirical Prediction Interval
EPS	Ensemble Prediction System
EQD	Extreme Quantile Deviation
EV	Electric Vehicle
FA	Feature Ablation
FP	Feature Permutation
GAN	Generative Adversarial Network
Gaussian PI	Gaussian Prediction Interval
GEFCom2014	Global Energy Forecasting Competition 2014
GPR	Gaussian Process Regression
HPO	Hyperparameter Optimisation
IG	Integrated Gradients
IoT	Internet of Things
LR	Linear Regression
MAQD	Mean Absolute Quantile Deviation
MW	Mean Winkler
N-HiTS	Neural Hierarchical Interpolation for Time Series Forecasting
NGBoost	Natural Gradient Boosting
NN	Feed-Forward Neural Network
NNQF	Nearest Neighbour Quantile Filter
NWP	Numerical Weather Prediction
PDF	Probability Density Function

Abbreviation	Description
PI	Prediction Interval
PL	Pinball Loss
QRNN	Quantile Regression Neural Network
RF	Random Forest
RMSE	Root Mean Squared Error
SVS	Shapley Value Sampling
TFT	Temporal Fusion Transformer
TIGGE	The International Grand Global Ensemble
UQD	Upper Quantile Deviation
XAI	Explainable Artificial Intelligence
XGBoost	eXtreme Gradient Boosting

Part I

Introduction

1 Motivation & Contribution

In our data-driven society, time series data is recorded at an increasing rate. These recorded time series often contain patterns, such as trends, seasonality, or irregular cycles. Furthermore, many of these time series are influenced by exogenous factors, such as weather conditions, natural disasters, technological advancements, or advertising campaigns [20]. For example, the electricity generated from a photovoltaic panel exhibits a clear daily pattern, fluctuates regularly due to local cloud cover, contains a yearly pattern due to changing seasons, and may be adversely affected by dust build-up or technological degradation. Furthermore, the sales of summer clothes exhibit a clear yearly seasonality but can also noticeably spike, perhaps due to the success of a new advertising campaign.

Due to these patterns and the influence of exogenous variables, almost all time series exhibit a certain degree of predictability [21], [22]. Once anomalies in the time series have been accounted for [23], this predictability can be used to generate time series forecasts, which are often crucial for many downstream applications [24]. For example, electricity demand forecasts are crucial for successful dispatch planning, maintaining grid stability and mitigating congestion [17]. Similarly, traffic congestion predictions can enable efficient route planning and assist decision-makers with long-term design choices regarding infrastructure development. Due to their importance, such forecasts must be trustworthy, i.e. they must deliver reliable and accurate predictions regarding the future [25], [26].

However, any forecast contains an inherent component of uncertainty due to, for example, the component of randomness contained within any time series, uncertainty in the exogenous variables that influence the time series, or uncertainty regarding the accuracy of the forecasting model [20]. Ignoring this uncertainty may lead to misleading and inaccurate forecasts since a forecast may contain so much uncertainty that the single realisation is little better than a random guess. As a result, a trustworthy forecast should quantify this uncertainty [26], [27]. Furthermore, the uncertainty should be quantified by considering known sources of uncertainty. For example, many time series are affected by meteorological uncertainty [28] and, therefore, this meteorological uncertainty should be considered in the uncertainty quantification. Alternatively, every time series can be considered as a realisation of a probabilistic distribution [20], and this underlying distribution of the data could also be useful in quantifying the uncertainty. Unfortunately, this underlying distribution is unknown, and therefore, it cannot be directly used to quantify the uncertainty. Furthermore, the probabilistic forecast resulting from the uncertainty quantification should demonstrate properties desired by the end user. If, for example, the user requires a conservative forecast that will definitely include worse-case scenarios and thus overestimates the uncertainty, then the uncertainty quantification should be flexible and

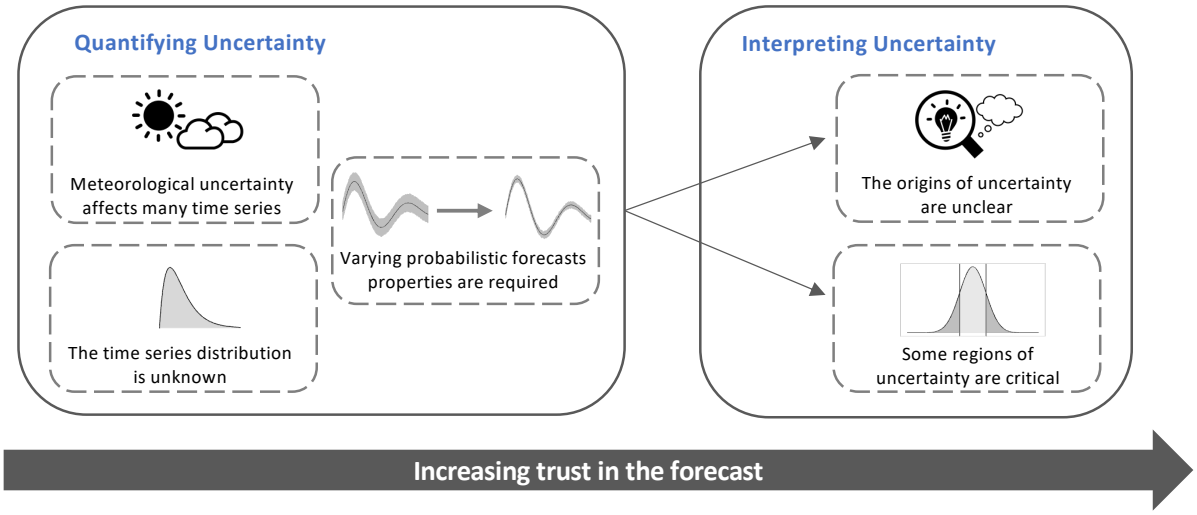


Figure 1.1.: The challenges associated with quantifying and interpreting uncertainty

designed to fulfil these requirements. As a result, the first step to increasing trust in a forecast is quantifying the associated uncertainty, ideally in a customised and flexible manner.

However, merely quantifying this uncertainty does not guarantee a trustworthy forecast since the resulting uncertainty must also be interpreted. Such a quantification of the uncertainty provides information regarding the amount of uncertainty present but fails to explain the origins of this uncertainty and why it occurs [29]. Without this explanation, it is impossible to determine when a forecast is erroneous or when it is accurately reacting to anomalous events in the data. Therefore, to further increase trust in a forecast, it should not only quantify the uncertainty but also explain the origins of this uncertainty [25], [29]. Furthermore, the uncertainty information should be presented in a manner that is useful for downstream applications making use of the forecast. In many applications, it is not the entire range of uncertainty that plays a critical role but rather certain critical regions of uncertainty [27]. For example, extreme temperatures play a critical role in crop farming, so the uncertainty regarding such extreme temperatures is most important. Therefore, the quantified uncertainty must be represented in a way that highlights the risk surrounding extreme temperatures. Similarly, energy systems models designed to determine viable future energy systems are affected by extreme weather scenarios, and, therefore, the representation of the quantified uncertainty should specifically focus on such events. Therefore, forecasts can only be trustworthy if a representation of the uncertainty quantification is created that is particularly useful for the considered application [27].

These two key steps in dealing with uncertainty, i.e. *quantifying* and *interpreting* uncertainty, along with the key challenges for each step are highlighted in Figure 1.1. Although quantifying uncertainty in general has been the focus of much research for many years, there has been no work specifically looking at linking meteorological uncertainty, using the unknown time series distribution, or customising the properties of probabilistic forecasts. Additionally, hardly any research focuses on interpreting uncertainty, specifically on explaining the origins of uncertainty or developing representations of this uncertainty that are useful for a considered application.

Table 1.1.: An overview of existing literature dealing with quantifying and interpreting uncertainty in time series. Whilst multiple papers consider two of the three aspects, none of the existing literature simultaneously considers quantifying, explaining and representing uncertainty.

Paper	Quantifying Uncertainty	Interpreting Uncertainty	
		Explaining	Representing
Abdar *et al.* [30]	✓	✗	✗
Ghahramani [31]	✓	✗	✗
Gneiting and Katzfuss [32]	✓	✗	✗
Kabir *et al.* [33]	✓	✗	✗
Lim and Zohren [34]	✓	✗	✗
Martino [35]	✓	✗	✗
Nowotarski and Weron [36]	✓	✗	✗
Zhang *et al.* [37]	✓	✗	✗
Li *et al.* [38]	✓	(✓)	✗
Kono *et al.* [39]	✓	✓	✗
Lim *et al.* [40]	✓	✓	✗
Petropoulos *et al.* [41]	✓	✓	✗
Wickstrøm *et al.* [42]	✓	✓	✗
Zhao *et al.* [43]	✓	✗	✓
Leffrang and Müller [44]	✗	✗	✓

Furthermore, no work considers both of these steps together. Therefore, the present dissertation takes a holistic view by quantifying and interpreting uncertainty in time series forecasts, aiming to increase trust in forecasts. In the remainder of this first chapter, we highlight the identified research gap in Section 1.1 by considering a high-level overview of existing work. In Section 1.2, we present our research questions and contributions before outlining the structure of the present dissertation in Section 1.3.

1.1 Research Gap

This section considers existing work that deals with quantifying and representing uncertainty in time series forecasts.[1] We present an overview of the identified related work in Table 1.1 and describe each of the considered aspects of uncertainty in more detail in the following. Based on the existing literature, we also highlight the research gap the present dissertation addresses.

Quantifying Uncertainty By far, the most commonly considered aspect of uncertainty is the quantification of uncertainty. Simple attempts to quantify uncertainty have existed for many years [35], and today, consensus surrounding the fundamental principles of such probabilistic forecasts exists in the literature [32]. Specifically, probabilistic forecasts come in different forms, including distribution forecasts, prediction interval forecasts, quantile forecasts and scenario forecasts [45]. Furthermore, these forecasts may assume a parametric probability distribution or be non-parametric, focusing on elements of the empirical distribution [32].[2] Based on these

[1] This section only presents an overarching overview of existing literature. We explicitly consider in-depth related work for each topic in individual chapters of the present dissertation.

[2] These principles of probabilistic forecasting are introduced in more detail in Section 2.3.

underlying principles and forms of probabilistic forecasts, Zhang *et al.* [37] and Nowotarski and Weron [36] provide an overview of different probabilistic forecasting methods in the context of renewable energy forecasting, focusing on differentiating between classical approaches and machine learning approaches, and categorising these approaches. However, with the recent trend towards machine learning methods, most current research regarding uncertainty quantification also focuses on such methods. Focusing not only on time series forecasting, Kabir *et al.* [33] survey neural network-based methods for uncertainty quantification, Ghahramani [31] provides an overview of probabilistic machine learning methods, and Abdar *et al.* [30] compare deep learning approaches for uncertainty quantification. Lim and Zohren [34] focus on deep learning for time series forecasting and survey multiple approaches to quantifying the uncertainty in such forecasts.

Interpreting Uncertainty With regards to explaining uncertainty, explainable machine learning methods such as Shapley values [46], Grad-Cam [47], and layerwise relevance propagation [48] are increasingly incorporated in computer vision and natural language processing tasks. However, such explainable methods have, up until now, only been sparingly applied to time series analysis, focusing almost exclusively on time series point forecasts [49]. However, Seuss [29] argues that explaining uncertainty could increase trust in a probabilistic forecast. Despite this argument, only a few papers have explicitly attempted to explain uncertainty in time series forecasting. Petropoulos *et al.* [41] attempt to differentiate between model, data, and parameter uncertainty to determine which source accounts for the most uncertainty in the resulting prediction. While promising, this explanation is still broad and not necessarily useful for increasing trust [41]. Kono *et al.* [39] attempt to explain uncertainty in time series forecasts by interpreting the time series forecast as a multi-class classification problem and assuming a multi-peaked predictive distribution. The continuous time series variables are discretised into multiple bins where the correct class is the bin to which the true value belongs [39]. Given this setting, Kono *et al.* [39] can apply explainable methods from the image analysis domain, namely Generative Contributive Mapping. Although interesting, this approach is based on a complex transformation of a forecasting task to a classification task and fails to natively consider time series forecasts and explain the uncertainty explicitly for this task. Similarly, Wickstrøm *et al.* [42] attempt to combine uncertainty and explainability by training an ensemble of deep neural networks that each generates a point forecast and outputs a relevance score for each time step. Time steps with high relevance scores from multiple models in the ensemble are used to explain the output, whilst the standard deviation between the ensembles is considered a measure of uncertainty [42]. This approach, however, does not explicitly explain uncertainty but merely combines explainability and uncertainty with an ensemble. Li *et al.* [38] integrate an automatic relevance determination network into a deep state space model to generate probabilistic forecasts. However, whilst this automatic relevance network provides explanations regarding the importance of each input feature for the forecast, it does not explicitly explain the uncertainty [38]. Only Lim *et al.* [40] directly consider how explanations generated by attention maps affect the uncertainty of the resulting forecasts as a brief side note in their evaluation. However, the explanations generated by Lim *et al.* [40], as well the explanations from all other methods identified here, are generated

for a single probabilistic forecasting model and rely on the characteristics of this model, e.g. the attention mechanism. Therefore, such methods cannot be easily applied to existing machine learning models that generate probabilistic forecasts.

Regarding representing uncertainty, only limited research focuses on how uncertainty can be represented in a way that benefits downstream applications. Although Joslyn and LeClerc [27] observe that the representation of uncertainty must fit the given task and the user's expertise, this aspect of uncertainty has been mostly disregarded in the ensuing years. Leffrang and Müller [44] compare different visualisations of uncertainty and the effect that these various visualisations have on the user's trust in the forecast. Zhao *et al.* [43] focus on creating customised *cost-orientated* representations of probabilistic forecasts designed to minimise the operation cost of downstream applications specifically. However, their approach is limited to a single type or probabilistic forecast, namely a prediction interval [43]. Although different approaches for representing uncertainty are considered in other applications, such as medical image synthesis [50], we identify no further work focusing on the representation of uncertainty in probabilistic time series forecasting.

Identified Research Gap Although we identify work focusing on both quantifying and interpreting uncertainty individually, to the best our knowledge, no existing research focuses on these aspects in a holistic manner whilst considering trustworthiness. Furthermore, the identified work on quantifying uncertainty does not explicitly consider meteorological uncertainty, the underlying distribution of the data, or the generation of probabilistic forecasts with customised properties. Additionally, we identify a few papers that consider quantifying and interpreting uncertainty. However, regarding interpreting uncertainty, we do not identify any papers that attempt to explain the origins of uncertainty in probabilistic forecasts and create representations that are useful for downstream applications. Furthermore, none of the identified papers considers quantifying, explaining, and representing uncertainty simultaneously. Finally, most existing literature fails to address the question of trustworthiness and how this is related to quantifying and interpreting uncertainty in a probabilistic forecast.

1.2 Research Questions & Contributions

In the present dissertation, we aim to take a holistic view by quantifying and interpreting uncertainty in time series forecasts to generate trustworthy forecasts. We address these aspects of uncertainty in time series forecasts by answering five research questions. In the following we present these research questions and highlight how the present dissertation contributes to answering them.

Research Question 1: Can we use the meteorological uncertainty from weather forecasts to generate probabilistic forecasts for time series affected by meteorological conditions?

Meteorological variables, often in the form of weather forecasts, are one set of exogenous variables that affect many time series. Such time series include, for example, wind power production, which is closely linked to wind speed and direction, electricity demand, which is influenced by temperature, traffic congestion, which often increases with poor visibility and snow, and retail sales for products that are weather dependent. However, these meteorological time series are representatives of a complex and uncertain meteorological system. Therefore, any weather forecast itself is characterised by uncertainty. Unfortunately, existing probabilistic forecasting methods that only consider point weather forecasts as inputs fail to link this meteorological uncertainty with the uncertainty in the resulting forecast. Therefore, to address this challenge, we investigate methods for explicitly linking the uncertainty from meteorological time series to the uncertainty in the resulting probabilistic forecast. Therefore, one contribution of the present dissertation is the proposition and evaluation of multiple post-processing strategies to enable the linkage of meteorological uncertainty explicitly.

Research Question 2: Can the uncertainty information contained in the unknown distribution of time series data be used to generate probabilistic forecasts?

Many time series contain patterns such as trends, seasonality, or cycles. The information regarding these patterns and their associated uncertainty is implicitly contained in the underlying data distribution of the time series. Unfortunately, this data distribution is unknown and cannot be directly used to generate probabilistic forecasts. Therefore, to address this challenge, we propose a novel approach that maps the unknown distribution to a known and tractable distribution, which can be used to generate probabilistic forecasts from arbitrary point forecasts. This approach forms another contribution of the present dissertation.

Research Question 3: Is it possible to customise the properties of a probabilistic forecast, without retraining or developing an alternative forecasting model?

Probabilistic forecasts are almost always generated for use in a certain application, and these applications often require specific forecast properties. Unfortunately, existing probabilistic forecasting methods can only generate probabilistic forecasts with customised properties if they are retrained with different loss functions or if alternative models are developed. Therefore, we propose extending our probabilistic forecasting method that uses the underlying distribution of the data with automated hyperparameter optimisation. This extension enables the generation of probabilistic forecasts whose properties can be customised according to a given probabilistic loss function without retraining the underlying forecasting model.

Research Question 4: Can existing Explainable Artificial Intelligence (XAI) methods be applied to explain the origins of uncertainty in a probabilistic forecast?

The fourth research question deals with whether the origins of uncertainty in a probabilistic forecast can be determined and explained. Although a probabilistic forecast quantifies the associated uncertainty, it does not explain the origins of this uncertainty or identify factors

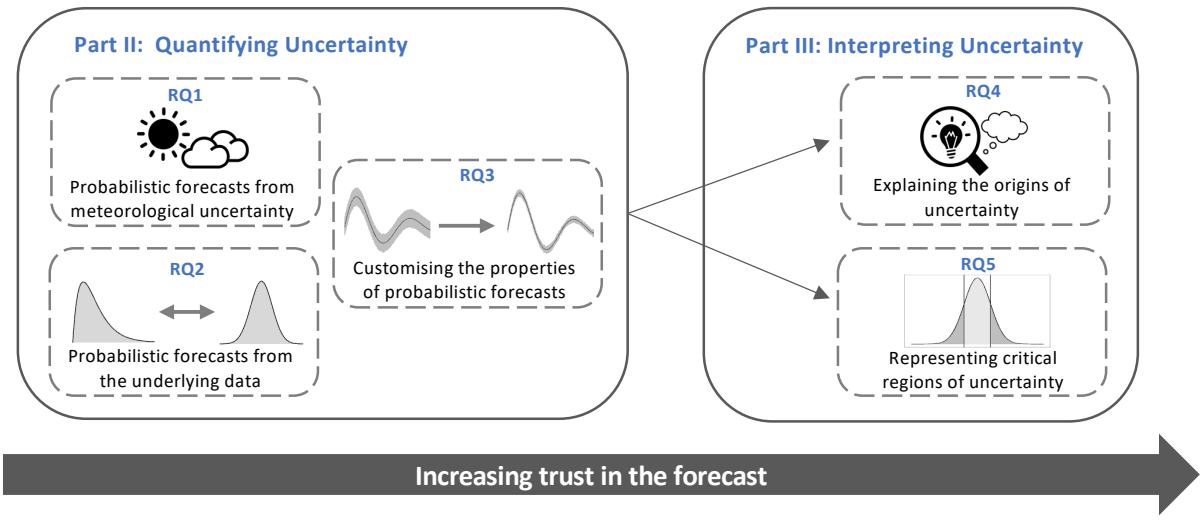


Figure 1.2.: A graphical overview of the content of the present dissertation.

leading to this uncertainty. Without these explanations, trusting the resulting forecast may still be difficult. For example, if the forecast fails to provide a plausible reason for the present uncertainty, it is impossible to determine whether unusual forecasts are erroneous or actually due to underlying factors that may not be easy to observe. Therefore, we present an approach that uses existing XAI methods to explain the origins of uncertainty in probabilistic forecasts. This approach, along with an evaluation on multiple data sets, is a further contribution of the present dissertation.

Research Question 5: How can we represent the quantified uncertainty in a way that is beneficial for downstream applications?

The final research question focuses on how to make probabilistic forecasts trustworthy for downstream applications. Many of these downstream applications struggle with probabilistic forecasts because certain regions of uncertainty are more critical than others for the application, and the probabilistic forecast does not explicitly highlight these regions. Therefore, with a focus on mobility applications, we aim to identify representations of these critical regions of uncertainty that are helpful for the downstream application. As a result, the final contribution of the present dissertation is the presentation and comparison of multiple representations of uncertainty, designed to highlight critical regions of uncertainty for mobility applications specifically.

1.3 Outline

In this chapter, we motivated the importance of uncertainty for increasing trust in forecasts, provided an overview of existing work, and introduced our research questions and contributions. In the following chapter, which concludes Part I, we introduce the theoretical background for the present thesis including the foundations for time series, uncertainty in time series, and

probabilistic forecasting. The rest of the dissertation is structured into two main parts illustrated in Figure 1.2 to address the five research questions related to quantifying and interpreting uncertainty. In Part II, we address quantifying uncertainty and answer **RQ1**-**RQ3**. We continue in Part III, focusing on interpreting uncertainty and thereby addressing **RQ4** and **RQ5**. In the final part of the present dissertation, Part IV, we discuss the dissertation as a whole and how each of our contributions contributes to increasing trust in forecasts. Finally, we summarise our results and contributions and provide an outlook for future work.

2 Foundations

This dissertation deals with uncertainty in time series forecasting. Therefore, in this chapter, we first present some background information on time series in Section 2.1 before discussing uncertainty in time series forecasting in Section 2.2. Furthermore, we provide theoretical background for probabilistic time series forecasting in Section 2.3.

2.1 Time Series

Time series are the fundamental building block for the present dissertation. Therefore, in this section, we define time series and introduce the notation we use for the remainder of the dissertation before briefly highlighting some key time series characteristics.

2.1.1 Definition and Notation

A time series is an ordered sequence of observations, where this order is determined by time [51]. Formally, we define a time series $(\mathcal{Y}_t)_{t\in T} = (y_1, y_2, \ldots, y_T)$ as a realisation of the stochastic process $\{Y_t\}_{t\in T}$ [52]. Hereby, we consider a stochastic process as a collection of random variables on a common probability space and indexed by a given mathematical set T [53]–[55]. In the present dissertation, we focus on stochastic processes with real-valued realisations and consider the index set T as a set of time values measured at discrete intervals [56]. For such a stochastic process, each of the random variables Y_t is distributed according to a Probability Density Function (PDF) $f(Y_t)$ with the associated Cumulative Distribution Function (CDF) $F(Y_t)$ [52]. We use $\sim$ to denote *distributed according to*, and therefore represent a random variable distributed according to $f(Y_t)$ as $Y_t \sim f(Y_t)$. Given this definition, a time series $(\mathcal{Y}_t)_{t\in T}$ can be considered as a realisation of a T-dimensional random variable [57], i.e. a realisation of

$$(Y_1, Y_2, \ldots, Y_T) \sim f(Y_1, Y_2, \ldots, Y_T), \tag{2.1}$$

which is distributed according to the T-dimensional PDF $f(Y_1, Y_2, \ldots, Y_T)$, and associated CDF $F(Y_1, Y_2, \ldots, Y_T)$ [55], [57].

In the present dissertation, we use vector notation to simplify expressions. Therefore, we define a time series via

$$\mathbf{y} = (\mathcal{Y}_t)_{t\in T} = (y_1, y_2, \ldots, y_T), \tag{2.2}$$

with the default length T. Hereby, the time series $\mathbf{y}$ is a realisation of a T-dimensional random variable

$$\mathbf{Y} \sim f_{\mathbf{Y}}(\mathbf{Y}), \tag{2.3}$$

with $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_T)$ the T-dimensional random variable distributed with the PDF $f_{\mathbf{Y}}(\mathbf{Y})$ and associated CDF $F_{\mathbf{Y}}(\mathbf{Y})$. Furthermore, we often consider a subsection of a time series, i.e.

$$\mathbf{y}_{t+H} = (y_{t+1}, y_{t+2}, \ldots, y_{t+H}), \tag{2.4}$$

which refers to the next H values of the time series from point t or

$$\mathbf{y}_{t-P} = (y_{t-P+1}, y_{t-P+2}, \ldots, y_{t-1}, y_t), \tag{2.5}$$

which refers to the previous P values of the time series up until the current point in time t. This notation serves as the basis for the present dissertation, and further notation specific to each contribution is introduced in the corresponding chapter.

2.1.2 Time Series Characteristics

Many time series exhibit characteristics such as trend, seasonality, and cyclic behaviour [20]. These characteristics are shown in Figure 2.1, and following the definitions from Hyndman and Athanasopoulos [20], we briefly describe them in the following.

First, a trend occurs when there is a noticeable long-term upward or downward tendency to the time series. Such trends may not necessarily be linear and can also change direction at some point in time. Trends are typically observed in many different types of time series, for example, economic indicators such as gross domestic product time series, population growth time series, and temperature time series, especially due to the effects of global warming.

Second, seasonality occurs in a time series when this time series is affected by calendar information. Seasonality can involve daily patterns, weekly patterns, or, most commonly, yearly patterns. Importantly, seasonalities are always of a fixed and known frequency. Many time series, such as product sales and electricity consumption, exhibit seasonalities. In this case, product sales often contain a clear seasonal pattern due to holidays and festive seasons. Meanwhile, electricity consumption exhibits a daily, weekly, and yearly seasonality due to weather conditions affecting heating and cooling demand.

Often confused with seasonality, cyclic behaviour occurs when a time series contains a pattern or rises and falls that is not fixed or regular. More specifically, the rises and falls must be visible, however, they are not of a fixed frequency and cannot be explained by seasonal factors. The so-called business cycle is the most common example of cyclic behaviour in time series. These business cycles consist of alternating periods of expansion (economic growth) and contraction (recession) and typically span several years.

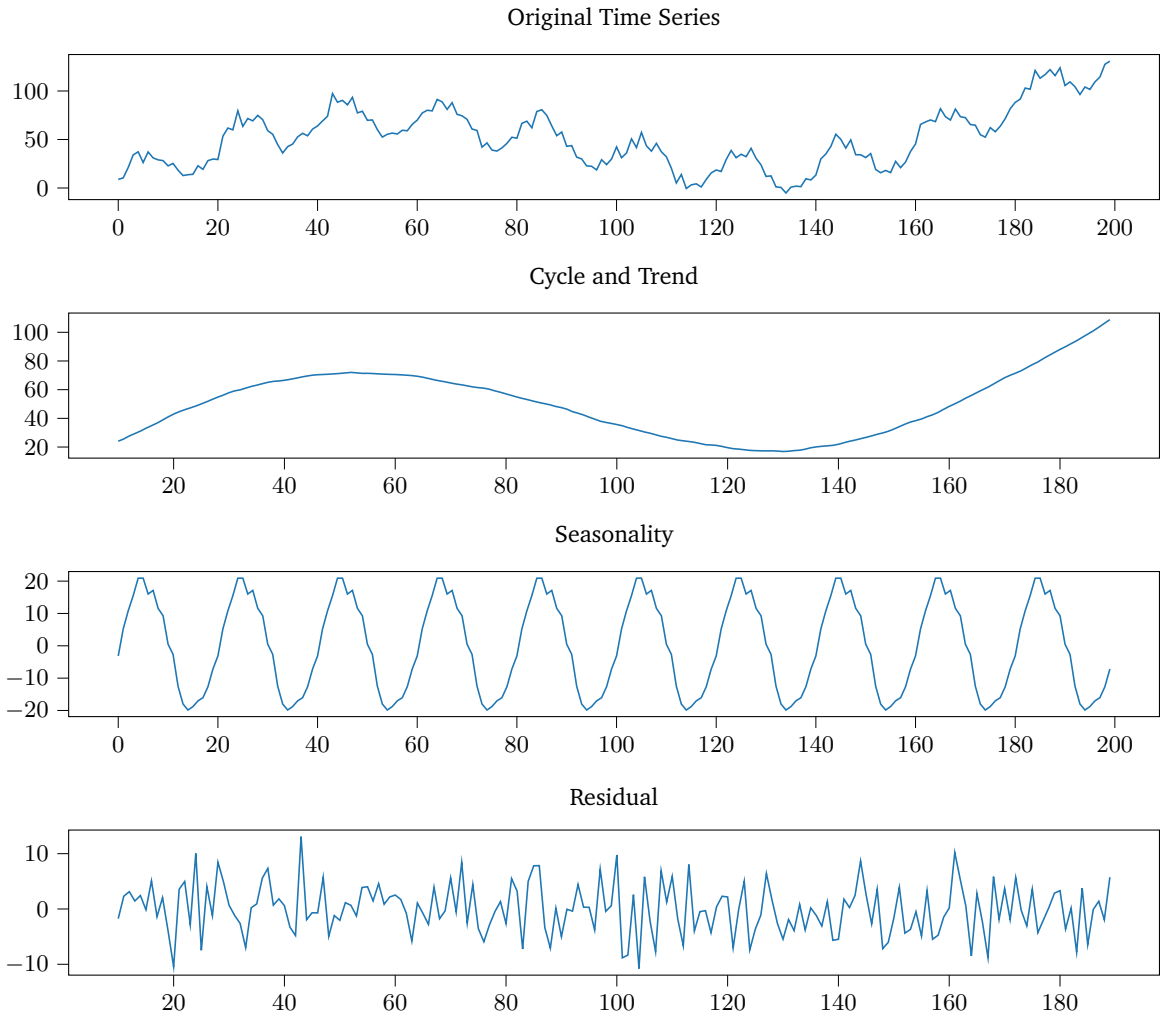


Figure 2.1.: A time series with varying characteristics. The time series contains an upward trend and regular seasonality. Furthermore, the time series seems to contain a cycle, however, this is difficult to confirm without further observations. As expected, the residual of the time series cannot be easily explained by a trend or seasonal pattern, and exhibits the characteristics of random noise.

When analysing time series, it is possible to perform a time series decomposition to isolate the trend and seasonal components of the time series [20]. This decomposition may be useful, since it can highlight aspects of the time series that are difficult to predict and assist with model development. More specifically, to perform a time series decomposition, the original time series $(\mathcal{Y}_t)_{t\in T}$, can be written as:

$$(\mathcal{Y}_t)_{t\in T} = (\mathcal{T}_t)_{t\in T} \odot (\mathcal{S}_t)_{t\in T} \odot (\mathcal{C}_t)_{t\in T} \odot (\mathcal{R}_t)_{t\in T}, \tag{2.6}$$

where $(\mathcal{T}_t)_{t\in T}$ is the trend component, $(\mathcal{S}_t)_{t\in T}$ the season component, $(\mathcal{C}_t)_{t\in T}$ the cyclic component, and $\odot$ the operator used for the decomposition. Most commonly we consider an additive decomposition (i.e., $\odot = +$), however a multiplicative decomposition (i.e.$\odot = \cdot$) is also possible. After performing the time series decomposition, the remaining component $(\mathcal{R}_t)_{t\in T}$ is the so-called residual. This is the component that cannot be easily explained by a trend, cycle, or seasonal

patterns, and is often little more than random noise. Therefore, the residual is usually the most unpredictable component of the time series.

2.2 Uncertainty in Time Series Forecasting

A time series forecast contains, per definition, an inherent component of uncertainty [52]. This is because each time series is simply a single realisation of a stochastic process [55], and therefore, predicting any future realisations involves dealing with the stochastic uncertainty of this process. However, other factors besides this stochastic uncertainty also increase the uncertainty in any time series forecast. Therefore, in this section, we briefly define the main types of uncertainty relevant to time series forecasting before describing some further possible sources of uncertainty.

2.2.1 Types of Uncertainty

Uncertainty exists in many domains and is broadly defined as a state of limited knowledge, where it is impossible to exactly describe future outcomes [58]. Given this broad definition, many attempts have been made to classify types of uncertainty, including classification according to *effect and probability* and *data, component, and structure* [59], or by deriving taxonomies of uncertainty [60]. However, in terms of time series forecasting, two types of uncertainty are commonly considered: *aleatoric* uncertainty and *epistemic* uncertainty [61]–[63]. In the following, based on the description from Guo *et al.* [61], we introduce both types of uncertainty.

Aleatoric uncertainty refers to the inherent randomness of the data that cannot be explained away. As a result, regardless of the knowledge available, aleatoric uncertainty cannot be reduced. As a result, the only way to cope with aleatoric uncertainty is to factor it in when making decisions. Commonly, the aleatoric uncertainty is assumed to be either homoscedastic, i.e. constant, or heteroscedastic, i.e. variable. Typical examples of Aleatoric uncertainty include noisy data and underlying complex and highly stochastic processes, such as our meteorological system, which are inherently uncertain.

Unlike aleatoric uncertainty, epistemic uncertainty is a subjective uncertainty that results from a lack of knowledge. This lack of knowledge may be, for example, due to incomplete information, such as missing or ignored exogenous inputs to a model, an insufficient amount of data to prove a hypothesis, or limitations in our understanding of a model or system. Epistemic uncertainty can typically be reduced. To reduce such uncertainty, our knowledge must increase. Typically, we increase our knowledge by, for example, expanding the data available for training our models, including additional input variables, carrying out further research to better understand the underlying system, refining our models to reflect system behaviour more accurately, or taking advantage of new technologies.

Importantly, even with advanced technologies capable of reducing epistemic uncertainty, a component of aleatoric uncertainty will always remain. Therefore, forecasts that quantify this uncertainty will always be important, even if the epistemic component can be reduced.

2.2.2 Sources of Uncertainty

There are many sources of uncertainty in time series analysis. However, for the present dissertation, we summarise some main sources of uncertainty identified by Guo *et al.* [61], Linkov and Burmistrov [64], Walker *et al.* [65], Brugnach *et al.* [66], Bauer *et al.* [67] and Zimmermann [68] in the following.

Chaotic System A system may display chaotic and unpredictable behaviour over time. Additionally, a time series with a clear underlying pattern could contain large amounts of statistical noise, which is random and unpredictable, per definition. Therefore, uncertainty may be present in a forecast simply because the underlying system is chaotic and does not follow a set pattern.

Data Collection Uncertainty may be caused by the data collection process. This may be, for example, because not enough data is available to understand the time series to be forecast accurately. On the other hand, this may be because the collected data does not represent all aspects of the underlying process. If all data was collected within a small time interval, for example, then this data doesn't provide any information on the behaviour or the time series outside this time interval. Therefore, uncertainty increases for all periods outside the considered interval.

Model Development Any model used to forecast a time series is an abstraction of reality. In the development of this model, assumptions regarding the underlying process are made, and a certain structure for the model is developed. Due to this abstraction of reality and the associated assumptions, uncertainty is automatically included in the forecast.

Parameter Estimation After a model structure is determined, the model's parameters are estimated. This process, however, also includes uncertainty since this estimation is based on a training process that may be stochastic by nature, which may lead to different locally optimal parameters being identified in each training run [69], [70]. Therefore, in addition to uncertainty from the model structure, uncertainty is also included during the parameter estimation process.

Uncertain Exogenous Factors Many time series are affected by exogenous factors such as weather conditions. These exogenous factors are in themselves often characterised by uncertain or unpredictable behaviour. As a result, the uncertainty within these exogenous factors also contributes to the uncertainty in the time series they influence.

Human Behaviour For many time series, especially in domains such as economics and mobility, human behaviour can be a major source of uncertainty. Human behaviour, since not always

rational, is notoriously difficult to predict, and therefore, whenever humans influence a time series, they automatically include uncertainty in this time series.

2.3 Probabilistic Forecasting

Probabilistic forecasts are designed to quantify the above-mentioned uncertainty in time series forecasts. Before we explicitly consider probabilistic forecasts, it is worth looking at the definition of a point forecast. Such a point forecast can be considered as the conditional expectation of the future course of the time series given all available information [45], i.e.

$$\hat{\mathbf{y}}_{t+H} = \mathbb{E}[\mathbf{Y}_{t+H} \mid g, \hat{\Theta}, \circ], \tag{2.7}$$

where $\hat{\mathbf{y}}_{t+H}$ is point H-step forecast, $\mathbf{Y}_{t+H}$ a H-dimensional random variable modelling a stochastic process over the period of the forecast horizon, g an arbitrary prediction model with estimated parameters $\hat{\Theta}$, and $\circ$ are further features considered for the forecast. Thereby, $\circ$ could include exogenous features for the forecast horizon $t + H$ or historical values $t - P$ from observations before the forecast origin. Furthermore, these features may be point features, such as a single weather forecast or probabilistic, such as a weather ensemble forecast. Although the resulting point forecast $\hat{\mathbf{y}}_{t+H}$ is an expected value that implicitly indicates the presence of variance and uncertainty, point forecasts fail to quantify this uncertainty. Therefore, probabilistic forecasts are designed to quantify the underlying uncertainty actively. In this section, we introduce some key properties of probabilistic forecasts, present the general forms of probabilistic forecasts, and introduce some strategies for evaluating probabilistic forecasts.

2.3.1 Properties of Probabilistic Forecasts

It is relatively simple to understand what characterises a good point forecast, i.e. the forecast value should be as close to the actual observation as possible. However, when considering probabilistic forecasts, determining desirable properties is not such a trivial task. As a result, much literature exists analysing the properties of probabilistic forecasts, namely Gneiting *et al.* [71], Gneiting and Katzfuss [32], and Pinson *et al.* [72]. Focusing predominantly on the definitions from Gneiting *et al.* [71] and Gneiting and Katzfuss [32], we briefly discuss three key properties of probabilistic forecasts in the following: calibration, dispersion, and sharpness.

The first key property of a probabilistic forecast is calibration. Statistical calibration can be considered as a measure of statistical compatibility between the observations and the forecasts. Ideally, it should not be possible to distinguish the observations from random draws of the predictive distributions, i.e. a well-calibrated forecast implies that the predicted probabilities match the observed frequencies of events over a large number of forecasts. For example, if a probabilistic forecast always suggests there is 70% chance of rain, then over a reasonable

observation period containing sufficient forecasts and observations, it should actually rain approximately 70% of the time.

The second property of a probabilistic forecast is dispersion. Dispersion refers to the spread or variability of the predicted probabilities assigned to different outcomes and is thus a joint property of forecasts and observations. A forecast with high dispersion implies a wide range of probabilities assigned to potential outcomes, indicating uncertainty or lack of confidence in the prediction. Conversely, low dispersion indicates a narrow range of probabilities, suggesting higher confidence in the forecast. It is important to note that calibration and dispersion are closely related and essentially measure the same aspect of a probabilistic forecast from different perspectives [32]. Whilst calibration focuses on the predicted probabilities matching the observed probabilities, dispersion focuses on the variation between these properties. Considering the rain example above, if the forecast is repeated for multiple days with different rain probabilities being forecasted each day, the forecast can be considered well-dispersed if this range of probabilities accurately reflects the true outcome. In this case, the forecast is automatically well-calibrated.

The final property considered here is sharpness, which considers the concentration or narrowness of the predicted probability distribution around the observed outcome. A sharp forecast implies that the assigned probabilities are concentrated near the observed outcome, reflecting higher confidence and precision. On the other hand, a less sharp forecast exhibits a broader probability distribution, indicating greater uncertainty and less precision in predicting the outcome.

Whilst calibration, dispersion and sharpness are all important properties in a probabilistic forecast, Gneiting and Katzfuss [32] argue that probabilistic forecasts should aim to maximise the sharpness of the predictive distribution, subject to calibration. This approach ensures that the resulting probabilistic forecast provides enough information to be useful whilst remaining calibrated. To illustrate this idea, we consider the example of forecasting the chance of rain. If the forecast solely focused on calibration, the probability of rain occurring would almost always be perfectly forecast. However, to ensure that extreme probabilities, such as 1%, are also accurately forecasted, the resulting probabilistic forecast would have wide prediction intervals. Making decisions based on such probabilistic forecasts would be difficult since they would almost always predict rain. If we were to only focus on sharpness, however, the resulting forecast would be overly concentrated and often underestimate the probability of rain. Making decisions based on these concentrated forecasts would also be difficult since it often rains when unexpected. Therefore, a trade-off is necessary where some calibration accuracy is sacrificed in order to generate a probabilistic forecast that is sharp enough to be still useful.

2.3.2 Forms of Probabilistic Forecasts

Many methods and models exist to create probabilistic forecasts [32]. However, four main forms of probabilistic forecasts are commonly applied across domains in the literature: distribution forecasts, quantile forecasts, prediction interval forecasts, and scenario forecasts, also known as ensemble forecasts [45]. These different forms of probabilistic forecasts are sketched in Figure 2.2

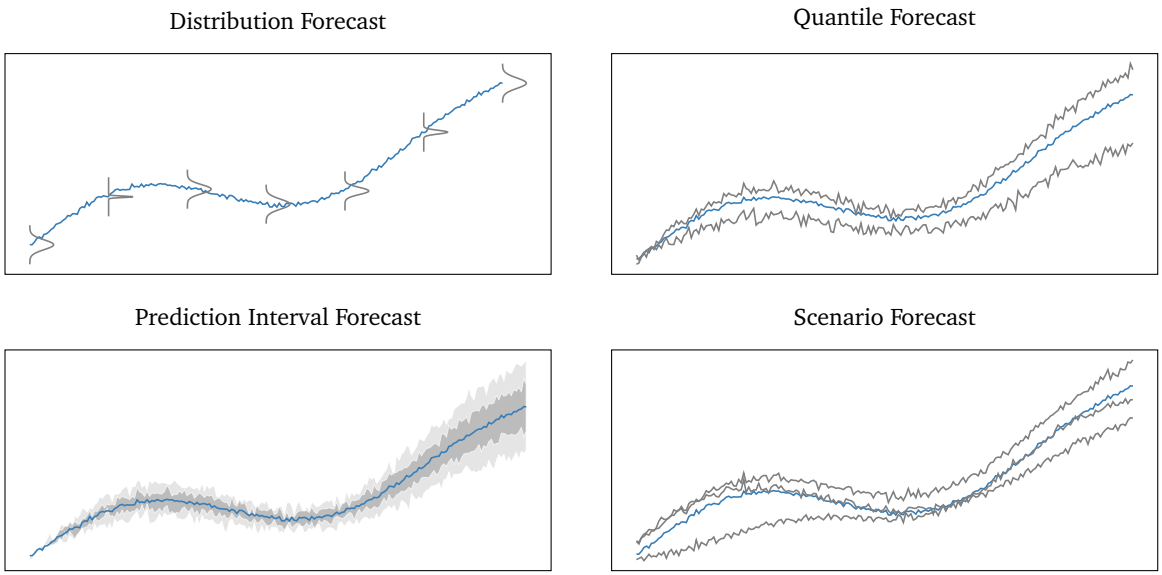


Figure 2.2.: An overview of the main forms of probabilistic forecasts. In a *distribution* forecast, a full CDF is forecast for each time step. Individual quantiles are combined to quantify the uncertainty in a *quantile* forecast. In a *prediction interval* forecast, a range of potential values for the ground truth is forecast. Finally, a *scenario* forecast generates multiple point predictions that, when combined, quantify the uncertainty in the forecast.

and in the remainder of this section, we introduce and explain each of these forms of probabilistic forecasts.

Distribution Forecast A distribution forecast predicts the full PDF or CDF for the forecast horizon, conditional on the information available. Therefore, a distributional forecast results in a H-dimensional predicted PDF $\hat{f}(\mathbf{y}_{t+H})$ or CDF $\hat{F}(\mathbf{y}_{t+H})$. Hereby, the predicted distribution may be parametric, i.e. it follows a known parametric distribution such as a Gaussian or Logarithmic distribution. In this case, the probabilistic forecast predicts the parameters of the parametric distribution. However, the predicted distribution may also be non-parametric if estimated based, for example, on Bernstein polynomials or histogram networks Schulz and Lerch [73]. A distribution forecast is the most general form of a probabilistic forecast, and it is possible to extract other forms, e.g., quantile forecasts or prediction interval forecasts, from the full distribution.

Quantile Forecast A quantile forecast $\hat{\mathbf{y}}^{(\alpha)}_{t+H}$, with nominal level α, is a point forecast with the probability α that the observation $\mathbf{y}_{t+H}$ is smaller than the quantile forecast $\hat{\mathbf{y}}^{(\alpha)}_{t+H}$ [45], i.e.

$$\mathbb{P}[\mathbf{y}_{t+H} \leq \hat{\mathbf{y}}^{(\alpha)}_{t+H} \mid g, \hat{\Theta}, \circ] = \alpha. \tag{2.8}$$

For example, with $\alpha = 0.5$, the probability of the observation being smaller than the quantile forecast is 50%, equivalent to forecasting the median. Whilst a quantile forecast by itself is a type of biased point forecast [15], we can forecast multiple quantiles and use this information to quantify the uncertainty. The quantification can be performed by generating prediction intervals

based on multiple quantiles or combining a large number of quantile predictions to convey information on the resulting PDF or CDF.

Prediction Interval Forecast A prediction interval forecast $\hat{I}_{t+H}^{(\beta)}$, with nominal coverage rate $1-\beta$, is a range of potential values with the probability $1-\beta$ of the observation $\mathbf{y}_{t+H}$ being contained in this range [45], i.e.

$$\mathbb{P}[\mathbf{y}_{t+H} \in \hat{I}_{t+H}^{(\beta)} \mid g, \hat{\Theta}, \circ] = 1-\beta. \tag{2.9}$$

Usually, these prediction intervals are formed by considering the range between two quantile forecasts, i.e.

$$\hat{I}_{t+H}^{(\beta)} = [\hat{\mathbf{y}}_{t+H}^{(\underline{\alpha})}, \hat{\mathbf{y}}_{t+H}^{(\bar{\alpha})}], \text{ where } \bar{\alpha} - \underline{\alpha} = 1-\beta, \tag{2.10}$$

where $\underline{\alpha}$ is the lower quantile, and $\bar{\alpha}$ the upper quantile. To ensure the prediction interval is centred on the PDF, we select symmetrical quantiles around the median, i.e.,

$$\underline{\alpha} = 1 - \bar{\alpha} = \beta/2. \tag{2.11}$$

As a result of this definition, ideal prediction intervals include $(1-\beta)\%$ of the observations.

Scenario and Ensemble Forecasts An alternative method of including uncertainty in probabilistic forecasts is a scenario or ensemble forecast [45], [74]. Such forecasts are based on different trajectories, where multiple options for the future are considered. These trajectories may be constructed based on different expectations for the future [45] or as a result of an ensemble of forecasting methods that represents the uncertainty in the forecast [74]. Although not governed by the same mathematical requirements as the other forms of probabilistic forecasts, scenarios or ensemble forecasts are an intuitive and simple way of quantifying the uncertainty in a time series forecast [74].

2.3.3 Evaluating Probabilistic Forecasts

Due to the properties of probabilistic forecasts described above, evaluating such forecasts is also challenging. As a result, much literature exists that addresses the evaluation question, including an information theory perspective from Roulston and Smith [75] and a theory specifically for ensembles from Anderson [76]. However, by far, the most comprehensive overview is provided by Tilmann Gneiting, with multiple works focusing on probabilistic forecasts, in general, [32], [71] and recently an additional paper solely focused on quantile forecasts [15]. Gneiting introduces multiple tools to evaluate probabilistic forecast properties, such as sharpness and calibration. Furthermore, a major contribution is so-called *proper scoring rules*, which provide summary measures of the predictive performance that allow for the joint assessment of calibration and sharpness in a probabilistic forecast [32]. These proper scoring rules are the foundation for most evaluations of probabilistic forecasts and have been implemented as a software package [77]. It

Table 2.1.: An overview of the evaluation metrics for probabilistic forecasts considered in the present dissertation. Some metrics are novel and introduced explicitly to cope with specific challenges considered in this dissertation. Note that metrics that evaluate calibration also automatically evaluate dispersion due to the close relationship of these two quantities.

Evaluation Metric	Calibration (Dispersion)	Sharpness	Novel
Continuous Ranked Probability Score [71], [78]	✓	✓	✗
Pinball Loss [15]	✓	✗	✗
Winkler Score [20], [79]	(✓)	✓	✗
Normalised Prediction Interval Width [20]	✗	✓	✗
Quantile Deviation [15]	✓	✗	✗
Coverage Rate Error [4]	✓	✗	✓
Extreme Quantile Deviation	✓	✗	✓
Upper Quantile Deviation	✓	✗	✓

is important to note that due to the close relationship between dispersion and calibration, proper scoring rules do not measure dispersion specifically, instead using calibration as a proxy.

Due to the thorough literature covering the evaluation of probabilistic forecasts and proper scoring rules, we refrain from a formal description in the present dissertation and refer to [15], [32], [71] for further information. Throughout the present dissertation, we consider multiple evaluation metrics for probabilistic forecasts. We present an overview of these metrics in Table 2.1 and introduce them briefly in the following.

Continuous Ranked Probability Score The Continuous Ranked Probability Score (CRPS) is a proper scoring rule that measures the calibration and sharpness of a predictive cumulative distribution function F [71], [78]. The CRPS is given by

$$\mathrm{CRPS}(F, y) = \int_{\mathbb{R}} \left(F(y) - \mathbb{1}\{y \leq z\}\right)^2 dz, \tag{2.12}$$

with the indicator function $\mathbb{1}\{y \leq z\}$ that is one if $y \leq z$ and otherwise zero. Furthermore, in the present dissertation, we always consider the average CRPS over all time steps $t = 1, \ldots, N$ in the test set

$$\mathrm{CRPS} = \frac{1}{N} \sum_{t=1}^{N} \mathrm{CRPS}(F_t, y_t). \tag{2.13}$$

Pinball Loss The Pinball Loss (PL) is a α-quantile consistent scoring rule that measures the error of each $\alpha-$quantile forecast by interpreting each of these forecasts as a point forecast [15].[1] Therefore, for a given quantile α the PL is given as

$$\mathcal{L}_{\mathrm{pinball},\alpha,i} = \begin{cases} (y_i - \hat{y}_i^{(\alpha)}) \cdot \alpha & \text{if } y_i \geq \hat{y}_i^{(\alpha)} \\ (\hat{y}_i^{(\alpha)} - y_i) \cdot (1 - \alpha) & \text{if } y_i < \hat{y}_i^{(\alpha)}, \end{cases} \tag{2.14}$$

[1] Note that the PL is not a proper scoring rule since it only considers an individual quantile and not the entire distribution. Therefore, it is not possible to evaluate calibration with the evaluation of a single quantile. However, evaluating multiple quantiles with the PL creates a discrete approximation of the CRPS, which is a proper scoring rule [15], [32].

where y_i is the observed value and $\hat{y}_i^{(\alpha)}$) is the quantile forecast for the quantile α. In the present dissertation, we often consider the mean PL across all quantiles and values in a test data set. This mean PL is thus defined as

$$\text{MPL} = \frac{1}{N \mid Q \mid} \sum_{\alpha \in Q} \sum_{i=1}^{N} \mathcal{L}_{\text{pinball},\alpha,i}, \tag{2.15}$$

for a set of given quantiles $\alpha \in Q$, and N observations in the test data set.

Winkler Score The third evaluation metric considered in the present dissertation is the Winkler score [79]. The Winkler score is designed to measure the quality of prediction intervals and is, therefore, unsuitable to evaluate a distribution or single quantile. As defined by [20], if the $100 \cdot (1-\alpha)\%$ prediction interval for observation i is given as $[\ell_{\alpha,i}, u_{\alpha,i}]$, then the Winkler score for the α-quantile is defined as

$$\text{W}_{\alpha,i} = \begin{cases} (u_{\alpha,i} - \ell_{\alpha,i}) + \frac{2}{\alpha}(\ell_{\alpha,i} - y_i) & \text{if } y_i < \ell_{\alpha,i} \\ (u_{\alpha,i} - \ell_{\alpha,i}) & \text{if } \ell_{\alpha,i} \leq y_i \leq u_{\alpha,i} \\ (u_{\alpha,i} - \ell_{\alpha,i}) + \frac{2}{\alpha}(y_i - u_{\alpha,i}) & \text{if } y_i > u_{\alpha,i}, \end{cases} \tag{2.16}$$

where y_i is the true value. Therefore, a Winkler score without any violations is simply the width of the prediction interval, whilst true values falling outside the prediction interval are penalised. Therefore, low Winkler scores suggest narrow but reasonably calibrated prediction intervals. We often consider the Mean Winkler (MW) score across all considered quantiles $\alpha \in Q$, defined as

$$\text{MW} = \frac{1}{n \mid Q \mid} \sum_{\alpha \in Q} \sum_{i=1}^{n} \text{W}_{\alpha,i}\,. \tag{2.17}$$

Normalised Prediction Interval Width The fourth evaluation metric considered is designed to evaluate the sharpness of the generated probabilistic forecasts. One way of measuring sharpness is the width of the prediction interval associated with a probabilistic forecast. However, since this width varies depending on the scale of the data set, it is important to normalise these widths. Therefore, to measure the sharpness of the probabilistic forecasts, we consider the normalised Mean β-PI Width ($\text{nMPI}(\beta)$), defined as

$$\text{nMPI}(\beta) = \frac{1}{\bar{y}} \left(\frac{1}{n} \sum_{i=1}^{n} \left| \hat{y}_i^{(\bar{\alpha})} - \hat{y}_i^{(\underline{\alpha})} \right| \right), \tag{2.18}$$

where $\hat{y}_i^{(\bar{\alpha})}$ is the predicted upper quantile, $\hat{y}_i^{(\underline{\alpha})}$ the predicted lower quantile for the forecast value $\hat{y}_i$, $\bar{y}$ the mean of the target time series, and $\bar{\alpha}$ and $\underline{\alpha}$ are as defined in Equation (2.11).

Quantile Deviation The fifth evaluation metric focuses on measuring the calibration of probabilistic forecasts. We consider the deviation of the forecast quantiles from the theoretical quantiles to analyse this calibration. We define the quantile deviation for the α-quantile as

$$\mathrm{QD}_\alpha = \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y_i \leq \hat{y}_i^{(\alpha)}\} \right) - \alpha, \tag{2.19}$$

where $\hat{y}_i^{(\alpha)}$ is the α-quantile forecast, y_i the true value, and $\mathbb{1}$ the indicator function. Ideally, QD_α should be zero for all values of α. However, a positive value indicates the quantile forecast overestimates the theoretical quantile, whilst a negative value indicates that the quantile forecast underestimates the theoretical quantile. We often visualise this deviation via quantile calibration plots [15]. To account for the total quantile deviation across all considered quantiles $\alpha \in Q$, we report the Mean Absolute Quantile Deviation (MAQD), i.e.

$$\mathrm{MAQD} = \frac{1}{\mid Q \mid} \sum_{\alpha \in Q} \left| \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y_i \leq \hat{y}_i^{(\alpha)}\} \right) - \alpha \right|. \tag{2.20}$$

Coverage Rate Error Whilst the MAQD accounts for calibration by considering each quantile individually, it doesn't consider the quality of the prediction intervals. Therefore we consider a novel evaluation metric, adapted from [16], which evaluates the Coverage Rate (CR) of the prediction intervals of the probabilistic forecast. The CR for the β-Prediction Interval (PI) is defined as the share of observed values that lie within the $\beta-$PI, i.e.

$$\mathrm{CR}_\beta = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1} \left\{ \hat{y}_i^{(\underline{\alpha})} < y_i \leq \hat{y}_i^{(\bar{\alpha})} \right\}, \tag{2.21}$$

where $\hat{y}_i^{(\underline{\alpha})}$ is the predicted lower quantile, $\hat{y}_i^{(\bar{\alpha})}$ the predicted upper quantile, and $\mathbb{1}$ an indicator function which is one if the observation is within the PI and zero otherwise. If the probabilistic forecast is perfect, then CR_β would be equal to $1 - \beta = \bar{\alpha} - \underline{\alpha}$. Therefore, to evaluate the quality of the CR, we consider the distance between the perfect and actual CRs, i.e. the Coverage Rate Error (CRE). For $\beta \in \mathcal{B}$, the Coverage Rate Error (CRE) is given by

$$\mathrm{CRE} = \sum_{\beta \in \mathcal{B}} \mid (1 - \beta) - \mathrm{CR}_\beta \mid . \tag{2.22}$$

Extreme Quantile Deviation Whilst the MAQD and CRE both consider the calibration of the entire forecast, for a given situation, the extreme regions of the probabilistic forecast may be interesting. Therefore, we introduce the Extreme Quantile Deviation (EQD) as a further loss metric, which is similar to the MAQD but only considers extreme quantiles, i.e. $\alpha \in \{0.01, 0.05, 0.95, 0.99\}$. By only considering extreme quantiles, the EQD places a higher weighting on extremes and will prefer probabilistic forecasts with wide prediction intervals that account for extreme regions of

uncertainty. Such a metric is particularly useful for customising the properties of probabilistic forecasts, as we discuss further in Chapter 5. Formally, we define the EQD as

$$\text{EQD} = \frac{1}{\mid Q_{\text{extreme}} \mid} \sum_{\alpha \in Q_{\text{extreme}}} \left| \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y_i \leq \hat{y}_i^{(\alpha)}\} \right) - \alpha \right|, \tag{2.23}$$

where $Q_{\text{extreme}} = \{0.01, 0.05, 0.95, 0.99\}$.

Upper Quantile Deviation The final metric considered in the present dissertation is a novel metric explicitly introduced for Chapter 5, where we customise the properties of probabilistic forecasts based on a given probabilistic loss function. The motivation for the Upper Quantile Deviation (UQD) is that, if the quantiles are symmetric around the median, it may be possible to customise the forecast to be close to optimal with regards to MAQD whilst only considering the upper part of these symmetrical quantiles. Such an assumption is reasonable if, for example, the considered distribution is Gaussian or similar. This metric would simplify the customisation procedure if successful since only half as many quantiles must be considered. Thereby, we define the UQD as

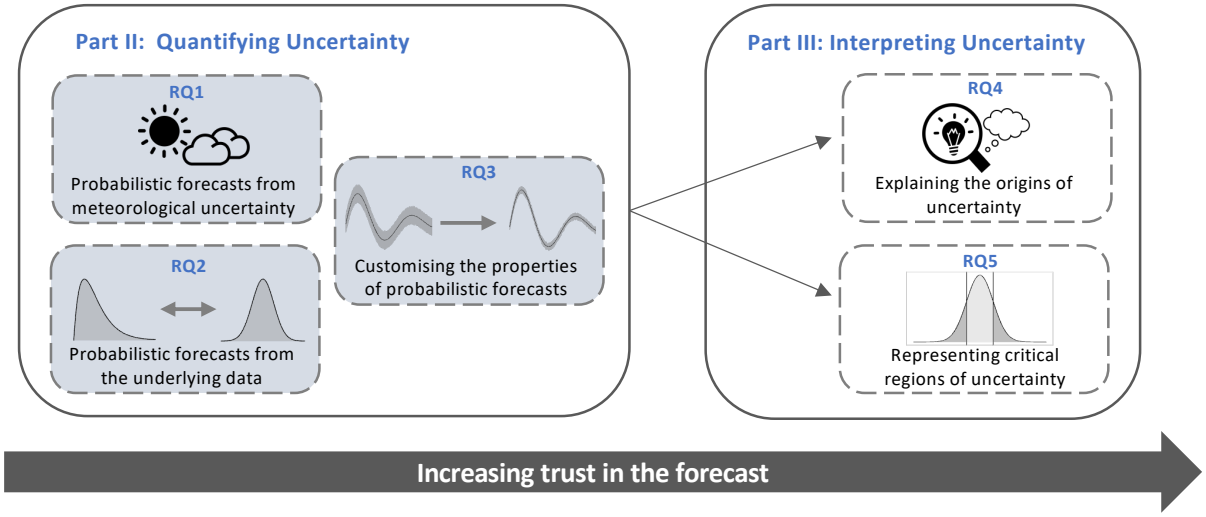
$$\text{UQD} = \frac{1}{\mid Q_{\text{upper}} \mid} \sum_{\alpha \in Q_{\text{upper}}} \left| \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{y_i \leq \hat{y}_i^{(\alpha)}\} \right) - \alpha \right|, \tag{2.24}$$

where $Q_{\text{upper}} = \{0.99, 0.95, 0.9, 0.85, 0.8, 0.75, 0.7\}$.

Part II

Quantifying Uncertainty

Overview of Part II



Part II of the dissertation focuses on quantifying the uncertainty in time series forecasts by generating probabilistic forecasts. Thereby we first answer **RQ1** in Chapter 3 by comparing post-processing techniques that enable the meteorological uncertainty in weather forecasts to be linked to the uncertainty in the target forecast. We compare four different post-processing techniques on multiple data sets for wind power forecasting. Our evaluation shows that such post-processing methods can indeed be used to link the meteorological uncertainty and that the most important post-processing step is always the target forecast.

We address **RQ2** in Chapter 4 by proposing an approach that generates probabilistic forecasts from arbitrary point forecasts by learning a known and tractable representation of the underlying data distribution. We show that a Conditional Invertible Neural Network (cINN) can be used to learn a mapping from the unknown realisation distribution to a known and tractable latent distribution and describe how this mapping can then be used to include uncertainty in the point forecast. Our results show that our approach generates high-quality probabilistic forecasts, comparable to or better than state-of-the-art benchmarks.

Finally, to answer **RQ3**, we show how the forecasting approach introduced in Chapter 4 can be extended with automated hyperparameter optimisation to customise the properties of the resulting probabilistic forecasts in Chapter 5. The properties of the generated probabilistic forecasts can be customised based on a given probabilistic loss metric and this customisation is possible without retraining the cINN or the applied base point forecaster.

3 Probabilistic Forecasts from Meteorological Uncertainty

The content of this chapter is based on:

K. Phipps *et al.*, "Evaluating ensemble post-processing for wind power forecasts", *Wind Energy*, vol. 25, no. 8, pp. 1379–1405, 2022. DOI: 10.1002/we.2736

K. Phipps, N. Ludwig, V. Hagenmeyer, and R. Mikut, "Potential of ensemble copula coupling for wind power forecasting", in *Proceedings 30. Workshop Computational Intelligence*, vol. 26, KIT Scientific Publishing, 2020, p. 87. DOI: 10.5445/IR/1000127955.

Meteorological conditions influence many time series. In some cases, this influence is clear, for example, the amount of wind power generated is proportional to the cube of the current wind speed, whilst weather conditions can be used as indicators for wind power ramps [80]. In other cases, the influence may be more subtle. Ice cream sales are volatile and affected by seasonal weather conditions [81] and have been shown to drop by up to 15% when the temperature drops [82]. Furthermore, in periods of noticeable temperature change, more seasonal clothing garments are sold [26] and increasing temperature fluctuations are altering typical clothes sales figures [83]. Whilst the degree of this influence varies, meteorological conditions are often considered important for the time series forecast. As a result, weather forecasts are often used as inputs in probabilistic forecasting models.

However, these weather forecasts are themselves uncertain since the meteorological system is chaotic and, as a result, difficult to predict [84]–[86]. As a result, meteorologists have developed complex Numerical Weather Prediction (NWP) models to model the physical relationships of the atmosphere and generate weather forecasts [85], [87]. Furthermore, these NWP models have been used to quantify the meteorological uncertainty via a so-called Ensemble Prediction System (EPS) [67], [88], [89]. An Ensemble Prediction System (EPS) is created by running the NWP model multiple times using slightly different model parameters each time. Therefore, the uncertainty contained in a EPS should, to a certain degree, account for the uncertainty in a time series that is influenced by meteorological conditions.

Unfortunately, this connection between meteorological uncertainty and the uncertainty in influenced time series is not accounted for when point weather forecasts are taken as inputs. Although such a connection could be established by connecting a NWP based EPS to the forecasting model, this poses a further challenge. The ensemble predictions are known to be biased and underdispersed and, therefore, it is standard practice to post-process EPSs before use [90]–[98].

Post-processing is used to alleviate systematic biases in the NWP model by calibrating the forecasts to past observations, and there is a large body of work discussing optimal post-processing methods for weather variables (see e.g. the review by Vannitsem *et al.* [99]). Therefore, to include meteorological uncertainty in a time series influenced by weather conditions, some form of post-processing must be performed.

However, generating time series forecasts includes an additional source of uncertainty – the epistemic uncertainty associated with the forecasting model. Therefore, there are three ways in which systematic biases can be present, making post-processing useful: in the first stage concerning the weather output from NWP models, in the second stage concerning the forecasting model, or both stages. As a result, determining at which state post-processing should be applied is crucial to successfully use EPSs to include meteorological uncertainty in a time series forecast. Previously, Pinson and Messner [100] explain the concept behind post-processing for time series forecasts by considering the example of wind power applications. They propose post-processing before the weather is used as input to the forecasting model and afterwards [100]. However, they do not evaluate or compare these different approaches. Furthermore, almost all work that considers ensemble post-processing focuses purely on post-processing the meteorological EPS and not any time series that are forecast based on these weather inputs. Such work focuses, for example, on post-processing wind speed EPSs [101]–[104] or comparing post-processing techniques for different weather variables [73], [105]–[107]. However, these papers all apply post-processing purely to the weather EPSs and do not consider the epistemic uncertainty resulting from using this EPS as an input to a forecasting model.

Therefore, in the present chapter, we evaluate four post-processing strategies to determine at which stage the post-processing is most beneficial in the forecasting process. We analyse whether one post-processing step can account for all previous biases by only post-processing the forecast target ensembles and compare this one-step strategy to post-processing only the weather ensembles, i.e. assuming the biases from the forecasting model are negligible. Lastly, we also compare one strategy where we post-process both weather and forecast target ensembles. We evaluate our strategies on the example of wind power forecasting using publicly available synthetic benchmark data and wind power data from two bidding zones in Sweden. The remainder of the present chapter is structured as follows. Section 3.1 introduces the theoretical background on the NWP model. We then describe the post-processing strategies in detail in Section 3.2, before presenting the experimental setting of wind power forecasting used to evaluate these strategies in Section 3.3. Further, we report the results of our evaluation of these strategies in Section 3.4. Section 3.5 discusses our approach before Section 3.6 concludes.

3.1 Numerical Weather Prediction Models

Atmospheric behaviour is chaotic and considered an unstable system which has finite, state-dependent limits of predictability [67], [84]. Numerical Weather Prediction (NWP) models describe and forecast this atmospheric behaviour, and with it the weather, through solving

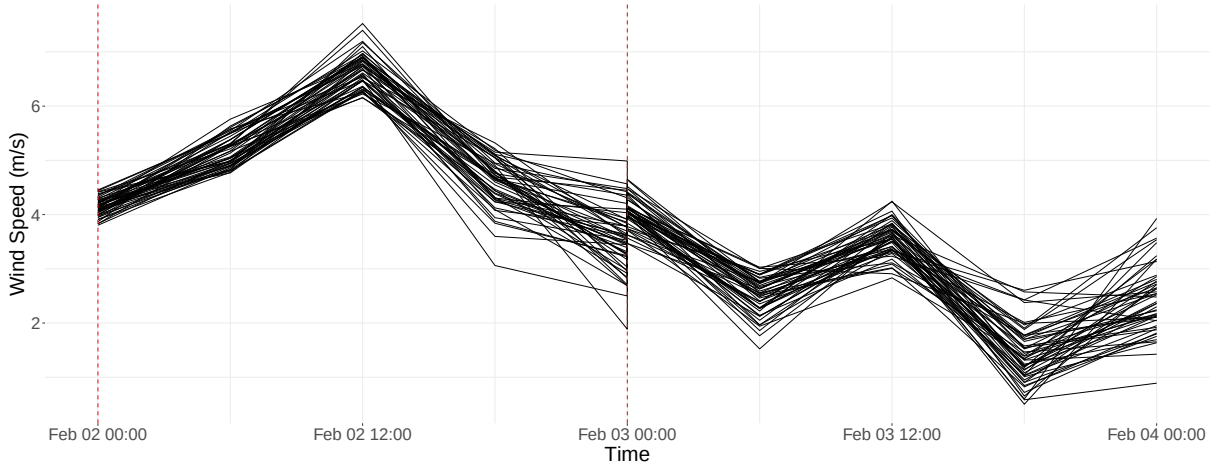


Figure 3.1.: An illustration of two consecutive ensemble forecast trajectories over a 24h forecast horizon for a wind speed EPS. The red dotted lines indicate the forecast origins. Close to the forecast origin the ensemble members all generate similar forecasts, however as the forecast horizon increases the ensemble members diverge. [1]

a system of non-linear differential equations starting with the current observed atmospheric conditions. As this current atmospheric state cannot be fully observed at any given point in time and space, there remains some uncertainty with regard to the initial conditions of the NWP models. However, forecasts of non-linear numerical models are highly sensitive to the given initial conditions, and initial errors grow during the forecast [67], [108]. Accounting for the uncertainty in the initial conditions is therefore crucial and nowadays quantified with the help of EPSs. EPSs generate ensemble forecast by running the NWP model several times with slightly different initial conditions, e.g. adding perturbations to the initial state. Hence, today's weather forecasts provide an inherent probabilistic uncertainty estimate in the form of ensembles of NWPs. Figure 3.1 exemplarily shows the ensemble forecast trajectories for a wind speed EPS over time for two consecutive forecast origins and a forecast horizon of one day. At each forecast origin, the EPS generates ensemble predictions for the specified forecast horizon. The forecast horizon describes the number of future time steps into which the weather is predicted. With increasing forecast horizon the trajectories diverge, resulting in an increased uncertainty associated with the chaotic behaviour of the non-linear weather system. For a more detailed overview of NWP and EPSs see e.g. Bauer *et al.* [67].

3.2 Ensemble Post-Processing Strategies

As the weather ensembles from the EPS are already known to be biased and underdispersed [90], different approaches exist in the meteorological literature to calibrate ensembles [99]. Two of the most common methods are Ensemble Model Output Statistics (EMOS) developed by Gneiting *et al.* [94] and Bayesian Model Averaging (BMA) introduced by Raftery *et al.* [109]. Both of these approaches have been successful in post-processing various weather ensembles (see for example, Javanshiri *et al.* [110] and Han *et al.* [111]), and the difference in performance

between the two models has been shown to be negligible [73], [111], [112]. However, EMOS is computationally far simpler than BMA [73], [113] and currently, operational implementations of post-processing at weather services focus almost exclusively on EMOS [93], [99]. For these reasons, the present chapter focuses on EMOS when comparing post-processing strategies. In the following, we describe the EMOS post-processing method in detail before introducing the post-processing strategies we evaluate in this chapter.

3.2.1 Ensemble Model Output Statistics

The EMOS method for ensemble post-processing, developed by Gneiting *et al.* [94], is based on non-homogeneous regressions. The standard EMOS approach is designed for individually distinguishable ensemble members. However, the present chapter uses ensemble members from the European Centre for Medium-Range Weather Forecasts (ECMWF). These ECMWF ensemble members are classified as *singular vector synoptic ensembles* and therefore exchangeable [88], [114]. Exchangeable ensembles represent equally likely future scenarios with no distinguishing features or ordering. Thus, they are ensembles with an invariant joint distribution function under permutation of the arguments [115]. This exchangeability implies that, for example, the ensemble labelled as the first ensemble member e_1 at forecast origin t is not related to the ensemble member with the same label at forecast origin $t+1$. Given exchangeable ensembles, EMOS expresses a univariate weather quantity W in terms of multiple linear regression on the M ensemble members, with equal weights for each exchangeable ensemble member [93], i.e.

$$W = a + b \cdot \sum_{i=1}^{M} e_i + \epsilon. \tag{3.1}$$

Hereby, $e_1, \ldots, e_M$ are exchangeable ensemble forecasts, a, b are regression coefficients, and ϵ is an error term with a zero mean. Given this point forecast, we can create a probability density function or probabilistic forecast by assuming a distribution. The parameters of the distribution can then be modelled given the mean and variance of the individual ensemble members. For example, assuming a normal distribution for the corresponding weather variable, we use the regression on W as an approximation for the mean μ and approximate the variance σ^2 as a linear function of the ensemble spread

$$\sigma^2 = c + d \cdot S^2 \tag{3.2}$$

where S^2 is the ensemble variance and c and d are non-negative coefficients. The resulting parametric model is then given by

$$W|e_1, \ldots e_M \sim \mathcal{N}\left(a + b \cdot \sum_{i=1}^{M} e_i, c + d \cdot S^2\right). \tag{3.3}$$

Different variables require a different choice of distribution, for example, wind speeds are restricted to positive values and exhibit a skewed distribution, and EMOS can be easily extended to these other distributions [94].

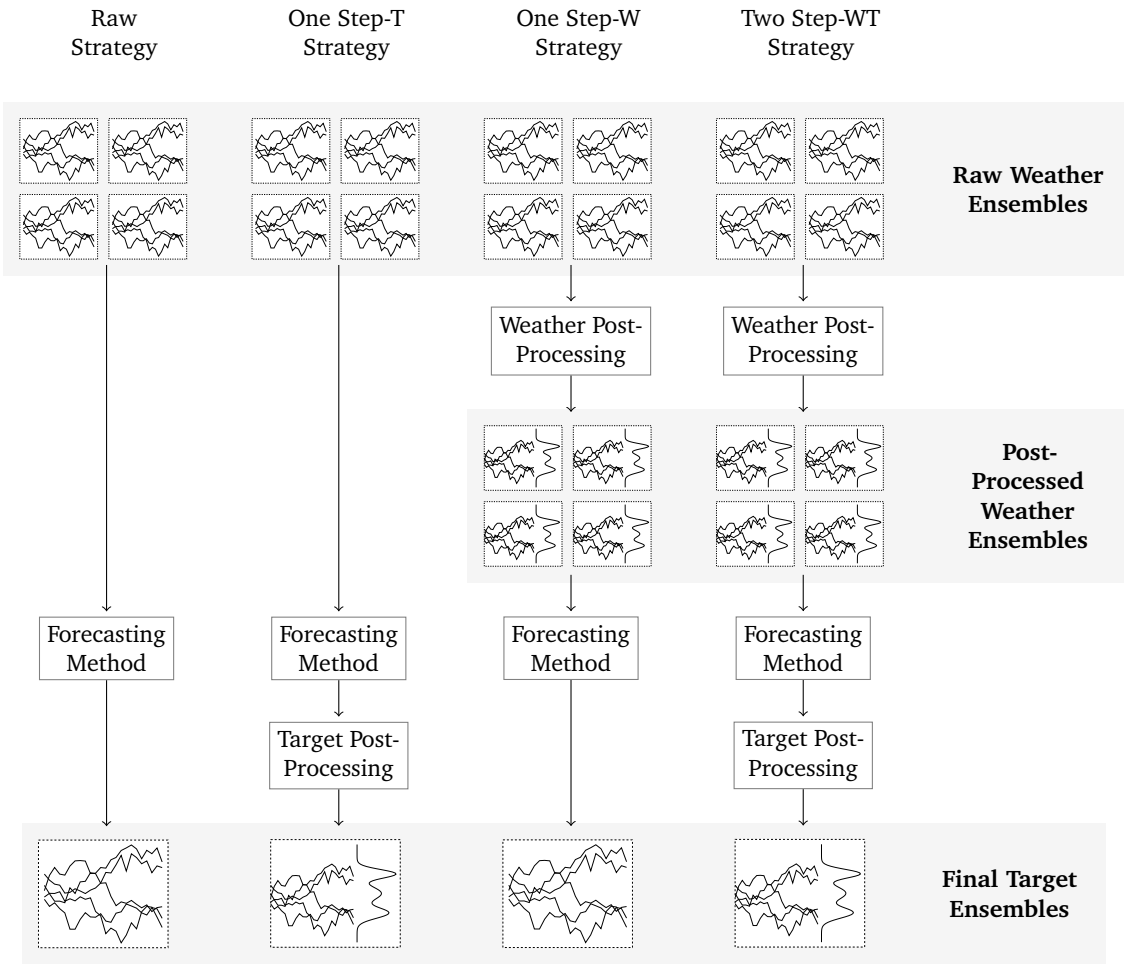


Figure 3.2.: An overview of the four post-processing strategies: No post-processing (Raw) where the input ensembles are directly used to generate the final target ensembles. Post-processing only the forecast target ensembles (*One Step-T*), where only the ensembles generated via the forecasting method are post-processed. Post-processing only the weather ensembles (*One Step-W*), where only the input ensembles are post-processed before being used as inputs into the forecasting method and the output is not post processed. Post-processing both weather and forecast target ensembles (*Two Step-WT*), where both the inputs and outputs are post-processed. Post-processed ensembles are identified in the via the distribution plotted on the right hand side of the ensemble plot, whilst raw ensembles do not contain this distribution.

3.2.2 Strategies

We now focus on the different post-processing strategies for probabilistic forecasts of time series that are affected by meteorological uncertainty. Figure 3.2 provides an overview of the forecasting process given the four different strategies, and we introduce each strategy in detail in the following.

Raw The *Raw* strategy serves as our benchmark and does not include any form of post-processing. All available M ensemble members from the EPS are fed through the same forecasting model individually, resulting in an ensemble of target forecasts with M members. This strategy assumes

that the probabilistic forecasting model can account for the bias in the EPS and, therefore, no post-processing is required.

One Step-T In the *One Step-T* strategy, post-processing is applied once for the final forecast target ensemble. This strategy assumes that post-processing the forecast target ensembles also accounts for the biases in the weather ensembles. We again start with the M raw ensemble members and feed them individually through the forecasting model. In contrast to the *Raw* strategy, the resulting forecast target ensemble is post-processed.

One Step-W In the third strategy, *One Step-W*, we post-process only the weather ensembles. Thus, instead of using the raw ensembles, as for the two previous strategies, we post-process the ensembles and then draw M independent samples from each of the resulting calibrated weather distributions. Whilst independent sampling ignores possible dependency structures between weather variables, we analysed these dependency structures and determined that restoring them via Ensemble Copula Coupling (ECC) has no noticeable effect on the results.[1] Therefore, we only consider a simple independent sampling method for this chapter. Furthermore, we select M samples to replicate the number of raw ensembles available. Since we consider multiple weather quantities as input (e.g. wind speed, temperature, etc.), we post-process each of these quantities separately, selecting a probability distribution that suits the considered weather quantity and applying EMOS. Overall, the *One Step-W* strategy assumes that all biases in the target time series forecast can be eliminated by accounting for the biases in the weather ensembles. Although this strategy is also classified as a one-step strategy, the number of post-processing steps depends on the number of different weather quantities considered as input since each of these weather quantities is a separate EPS that is post-processed separately.

Two Step-WT In the fourth strategy, *Two Step-WT*, both the weather ensembles and the ensemble of target forecasts are post-processed. This strategy relies on the assumption that neither of the one-step approaches can sufficiently account for all biases in the models and data, and the forecasts should be post-processed at all stages in the forecasting process. Thus, the two one-step strategies are coupled together, where we first apply the procedure described for *One Step-W* to the weather ensembles and then *One Step-T* to the ensemble of target forecasts.

3.3 Experimental Setting

We consider the use case of wind power forecasting to evaluate the four ensemble post-processing strategies. In this section, we introduce this setting by first describing the data used in Section 3.3.1 and Section 3.3.2. We then describe the forecasting models applied in Section 3.3.3,

[1] For detailed results and analysis see Phipps *et al.* [2].

Table 3.1.: Configuration parameters used to generate the wind power time series with the *Renewable.ninja* API. [1]

	Onshore Benchmark	**Offshore Benchmark**
Coordinates	51.0°N, 10.5°E	54.5°N, 6.0°E
Time Span	01-02-2017 – 31-08-2018	01-02-2017 – 31-08-2018
Capacity	130 MW	400 MW
Turbine Height	95m	90m
Turbine Type	Vestas V90 2000	Gamesa G128 5000
Similar Real Windpark	Windfeld Wangenheim-Hochheim-Ballstädt-Westhausen	BARD Offshore I

before discussing how EMOS was implemented in Section 3.3.4. Finally, we introduce the evaluation metrics considered in Section 3.3.5.[2]

3.3.1 Benchmark Data

Due to the lack of open source wind power data for specific wind parks, we generate benchmark wind power using the *Renewables.ninja* API[3]. Staffell and Pfenninger [116] verify the simulation and bias corrections implemented in the Renewables.ninja API are capable of reproducing accurate wind power time series. We replicate two real German wind power parks, one onshore and one offshore. Table 3.1 shows the parameters which we use, where we selected turbines with similar characteristics to those installed using the wind turbine database[4].

We use The International Grand Global Ensemble (TIGGE) archive[5] to access open-source ensemble weather data. TIGGE archive is a result of *The Observing System Research and Predictability Experiment*, which aimed to combine ensemble forecasts from leading forecast centres to improve probabilistic forecasting capabilities [117]. The archive includes a limited sample of the ECMWF ensembles from October 2006 until 2021. The limitations are placed on the available forecast horizons (only in steps of 6h instead of 3h in the licensed version of the EPS), the number of weather variables (e.g. wind speed and wind components are only available at the height of 10m and not also at 100m) and a reduced spatial resolution compared to the operational EPS. TIGGE archive is publicly accessible, and data can be downloaded through the MARS API. We use weather data for the same locations as the synthetically generated wind power data (see Table 3.1). We include the parameters temperature at two meters above ground, surface pressure, u-Component and v-Component of wind at 10m, and wind speed for the period from February 2017 until August 2018. The limited period of data available is due to damaged tapes in TIGGE archive, which affected all ensemble data, and the aforementioned period is the longest available with continuous weather data at the time of writing. For the ground truth historical weather data, we use the ERA5 reanalysis data [118]. This data is accessed via the Copernicus Climate Data

[2] Code to replicate results of this chapter is available via GitHub: `https://github.com/KIT-IAI/EvaluatingEnsemblePostProcessing`.
[3] `www.renewables.ninja`.
[4] `https://en.wind-turbine-models.com/turbines`
[5] `https://apps.ecmwf.int/datasets/data/tigge/`

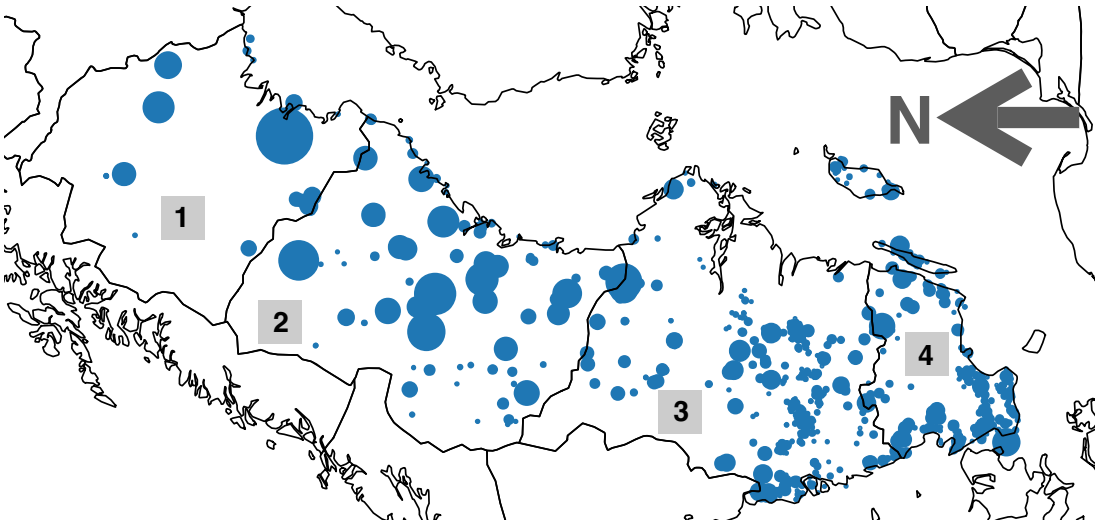


Figure 3.3.: A map of Sweden with the four bidding zones shown through the border lines. The blue circles indicate the distribution of wind turbines. The figure is adapted from Olauson and Bergkvist [119] and taken from Phipps *et al.* [1].

Store (CDS) API[6]. Here, the same locations and identical parameters are included. When working with the benchmark data set, we use data from the year 2017 for training and the remainder of the data (01.2018-08.2018) for evaluation.

3.3.2 Swedish Data Set

The second data set we use contains data from the Swedish electricity system. The Swedish electricity system is divided into four sub-areas or bidding zones [120]. The division of the electricity system into sub-areas has several purposes [121]. One purpose is to create regional price differences between the sub-areas. This is an incentive for cost-effective further development of the electricity system, as new power plants will be built where there is an electricity shortage. Another purpose is that state-owned Svenska Kraftnät, which operates the national grid, should receive indications about where the national grid needs to be strengthened to transfer enough electricity to the other sub-areas. A third purpose is to comply with EU legislation and facilitate the continued integration of the Swedish electricity system with the European electricity market.

In the present chapter, we focus on the area in bidding zones 3 and 4 for two reasons. Firstly, most wind power generation occurs in these two bidding zones (see Figure 3.3). Secondly, these are the two bidding zones in Sweden that are sometimes faced with a lack of electricity [121]. Whilst in northern Sweden, the supply of electricity is normally greater than the demand, transmission capacity between the north and south of Sweden is not always sufficient to transfer the demanded electricity, and this can lead to bottlenecks [121].

Weather data for bidding zones 3 and 4 consists of the ECMWF EPS (Molteni *et al.* [88]) and also the ERA5 reanalysis data C3S [118]. We use the EPS as the foundation for probabilistic

[6] `https://cds.climate.copernicus.eu`

Table 3.2.: A summary of the key characteristics of the weather data available for the use case in Sweden. [1]

Characteristic	Notes
Temporal Dimension	05.01.2015 – 31.08.2019
Spatial Dimension	11°E – 19.5°E 54°N – 62.5°N
Spatial Resolution	Grid resolution of 0.25°× 0.25°
Forecast Time	Forecasts for up to 24h ahead made at 00:00:00
Forecast Horizon	One step ahead forecasts for 3h, 6h, 9h, 12h, 15h, 21h and 24h
Weather Variables	100m u-component of wind, 100m v-component of wind, 100m wind speed, 2m temperature, surface pressure

Table 3.3.: A statistical summary of the wind power generation data for both benchmark data sets and for bidding zones 3 and 4 in Sweden. [1]

	Onshore Benchmark	Offshore Benchmark	Bidding Zone 3	Bidding Zone 4
Minimum	0.13 MW	1.2 MW	3 MW	2 MW
1st Quartile	11.31 MW	125.6 MW	278 MW	167 MW
Median	23.53 MW	236.8 MW	568 MW	352 MW
Mean	29.12 MW	223.9 MW	670.96 MW	447.55 MW
3rd Quartile	40.04 MW	333.2 MW	967 MW	664 MW
Maximum	128.83 MW	385.6 MW	2246 MW	1463 MW

forecasting methods and again use the ERA5 reanalysis data as the ground truth for the post-processing. Table 3.2 summarises the key aspects of the data. The weather data available is in a grid-based format. This means atmospheric models are used to create a NWP for certain geographical grid points on Earth. Since we are not considering a single wind park but looking at the aggregated wind power generation for each bidding zone, we considered all of these grid points as a weighted average. The weighted average is calculated as follows: Firstly, each data point is sorted into the appropriate bidding zone based on the geographical specifications (see Figure 3.3), and then a weighted average of every point is calculated. Due to a lack of accurate location data for various wind turbines in Sweden, a rudimentary weighted average method is used; areas with a high concentration of wind turbines are given double weighting, whilst those areas with a lower concentration only receive a standard weight. As seen in Figure 3.3, the doubly weighted areas include a central area in bidding zone 3 and the coastal areas in bidding zone 4.

The wind power generation data is available through the open-source transparency platform operated by the European Network of Transmission System Operators (ENTSO-E) [122]. The transparency platform provides aggregated onshore wind power generation data at an hourly resolution from 05.01.2009 until the present. This data is aggregated for each bidding zone in Sweden (zones 1-4), where we consider bidding zones 3 and 4 in the present chapter. We use data from 2015-2017 for the training of our models and then from 01.2018-08.2019 for the evaluation. Table 3.3 provides a statistical summary of the wind power generation data collected. The data collected is the raw wind power generation in Megawatt and is therefore affected by structural changes such as increased capacity, outages due to maintenance, and upgrades to wind turbines.

3.3.3 Forecasting Model

We aim to evaluate ensemble post-processing at different stages in the forecasting process, thus, we also need an appropriate wind power forecasting model. This forecasting model should represent the relationship between the different weather variables and the wind power and, in contrast to a manufacturers wind power curve model, can be estimated regardless of the turbine type and for aggregated wind power values (e.g. the sum of wind power from a large region with an unknown number of wind turbines). Since we focus on the comparison of different post-processing strategies and not raw forecast accuracy, we want the models to be simple, common in the literature, and lightweight with respect to parameter optimisation and computation time. Research has shown that post-processing generally improves regression results, even when deep learning methods are applied, see e.g. Kuleshov *et al.* [123], and therefore, we expect the post-processing performance to be similar independent of the model chosen. We, therefore, choose one linear model, namely a linear regression [124], as a simple benchmark model and one more complex non-linear model, namely a random forest [125].[7] In the following, we describe these two models and our forecasting strategy.

Linear Regression The linear regression model, which we use to forecast the wind power given the time series from different meteorological variables as input, can be described with

$$y_{t+h} = \beta_0 + \alpha y_{t+h-24} + \sum_{k=1}^{K} \beta_k W_{t+h}^k + \sum_{j=1}^{J} \gamma_j D_{t+h}^j + \varepsilon_{t+h}, \tag{3.4}$$

where y_t is the dependent variable, which in this case is the wind power, y_{t-24} is the actual wind power a day before, W^k are time series for different meteorological variables, including wind speed, the u-component of wind, the v-component of wind, surface pressure, and temperature, and D^j are variables containing temporal information, including the season, the month, and the year. The models are fitted for each forecast horizon $h = h_1, \ldots, h_H$ with $h_H \leq 24$, using actual historical weather data in order to describe the real relationship among the variables and remove any bias fitting on historical weather forecasts or ensembles could introduce. Each ensemble $e_1 \ldots e_M$ from the EPS is then used in a separate prediction run for each forecast horizon to generate an ensemble of wind power predictions with the previously fitted regression coefficients

$$\hat{y}_{t+h}(e_1, \ldots, e_M) = \hat{\beta}_0 + \hat{\alpha} y_{t+h-24} + \sum_{k=1}^{K} \hat{\beta}_k \widehat{W}_{t+h}^k(e_1, \ldots, e_M) + \sum_{j=1}^{J} \hat{\gamma}_j D_{t+h}^j + \hat{\varepsilon}_t. \tag{3.5}$$

Random Forest In order to better account for non-linear dependencies, we also implement a random forest. Random forests are a statistical learning method and are additionally to their ability to model non-linearity, also robust to errors when unnecessary predictors are included.

[7] To assess the robustness of our results we also consider an alternative linear regression model based on Zhang and Wang [126] and a simple artificial neural network as an alternative non-linear forecasting model. These models and the associated results are described in Appendix A.

Random forests create several de-correlated regression trees to form a collection of solutions. These de-correlated trees are fitted by selecting a subset of features at each candidate split. The final prediction is then the average over all regression trees. We use a random forest with 500 trees and the same input variables as for the linear regression to predict the wind power. Giving the random forest the full set of available input variables, specifically multiple weather features, allows it to automatically identify the most important features and generate accurate forecasts. The parameters of the random forest are, equivalently to the linear regression, estimated using the historical values, while the probabilistic ensemble forecasts are generated by running the forest on each ensemble member.

Forecasting Strategy The forecasting strategy is the same, regardless of the specific forecasting model. Given the forecast origin, we use weather forecasts $\widehat{W}^k_{t+h}$ from the EPS to predict the wind power $\hat{y}_{t+h}$ for each of the forecast horizons h (between 3h and 24h and determined by the temporal resolution of the EPS). Additionally, to the weather variables, we also include historical wind power generation from 24h before the prediction time and dummy features D^j_{t+h}, such as the time of day or month. Since the historical weather observations are implicitly included in the NWP model and the calibration process relies on these observations, we do not include historical weather information as a specific input for our prediction models. All forecasting models are implemented in R, and for the random forests, we use the *randomForest* package[8]. We apply the forecasting strategy explained above for every post-processing strategy shown in Section 3.2.2, such that the forecasting strategy does not influence our comparison.

3.3.4 EMOS Implementation

To estimate the EMOS coefficients, we follow Gneiting *et al.* [94] and use a minimum Continuous Ranked Probability Score (CRPS) (see Equation (2.12)) estimation based on the minimum contrast estimation approach. Gneiting *et al.* [94] show that the CRPS can be expressed as an analytical function, and the EMOS coefficients that minimise the CRPS can be found through the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.

To apply EMOS to our wind power ensemble, we assume a truncated normal distribution for the wind power using $\mathcal{N}^+_{[0,\infty)}(\mu, \sigma^2)$, with location $\mu = a + b\bar{e}$ and scale $\sigma^2 = c + dS^2$ as an affine function of the ensemble variance $S^2 = \frac{1}{M}\sum_{i=1}^{M}(e_i - \bar{e})^2$ [93]. We fit the ensemble of wind power forecasts to the historical wind power generation using a rolling EMOS approach, where the parameters are estimated every day based on the past 40 days. As an alternative, we also consider a gamma distribution for the wind power. For simplicity, this chapter focuses on the results using the truncated normal distribution, with the almost identical gamma distribution results presented in Appendix A.

Each weather variable is considered separately for the post-processing of the weather ensembles. For each of these weather variables, we use EMOS with a rolling training window to estimate

[8] `https://cran.r-project.org/web/packages/randomForest/randomForest.pdf`

the parameters for the appropriate distributions. We use the distributions as suggested by Gneiting [93], thus a normal distribution for temperature, a normal distribution for the u- and v-components of wind, and a truncated normal distribution or a gamma distribution for wind speed.[9] Given these distributions, we draw M independent random samples from each distribution to form calibrated weather ensemble forecasts. We then feed these sampled ensemble members through the forecasting model, resulting in an ensemble of wind power forecasts.

3.3.5 Evaluation Metrics

To evaluate the post-processing strategies, we consider the performance of the probabilistic forecasts resulting from the different strategies and their calibration. In this section, we briefly introduce the evaluation metrics used to assess this performance and calibration.

Performance To assess the forecast performance, we calculate the average CRPS over all time steps as defined in Equation (2.12). Furthermore, since we are mainly interested in a fair comparison of our post-processing strategies, we also calculate the CRPS Skill Score (CRPSS). This score measures the improvement in CRPS when compared to a given benchmark model. In our case, the CRPSS is calculated for all post-processing strategies with respect to the *Raw* strategy as the benchmark, i.e.

$$\mathrm{CRPSS} = \frac{\mathrm{CRPS}_{\mathrm{Raw}} - \mathrm{CRPS}_{\mathrm{Strategy}}}{\mathrm{CRPS}_{Raw}} \cdot 100, \tag{3.6}$$

where the CRPSS is given as a percentage, with positive results indicating an improvement.

Calibration The aim of post-processing and using EMOS is to obtain calibrated forecasts. To check whether a forecast is calibrated, we look at the *probability integral transform* (PIT) [127]. If F denotes a fixed, non-random predictive Cumulative Distribution Function (CDF) for an observation Y, the PIT is the random variable $Z_F = F(Y)$. If F is continuous and $Y \sim F$ then Z_F is standard uniform. Thus, ideally, the PIT from the given forecast is uniform. In the discrete case, where we do not have a CDF but instead multiple ensemble members, the PIT can be described by the verification rank histogram [128]. The verification rank histogram contains multiple bins formed from two ordered neighbouring ensemble members. Since the verifying ensembles are equally likely to fall within any of these bins in an ideal ensemble system, a rank histogram is also ideally uniform. Figure 3.4 sketches the key information in these histograms. If the post-processing is successful and the forecasts are calibrated, then we observe a uniformly distributed histogram. If the forecasts are underdispersed (i.e. they underestimate the true spread), then there are more observations in the outlying bins, and if the forecasts are overdispersed, there is more mass in the middle of the histogram (i.e. the forecast overestimates the true spread).

[9] Again, the gamma and truncated normal distribution delivered similar performance. For simplicity, we focus on the results from the truncated normal distribution in this chapter and report full results in Appendix A.

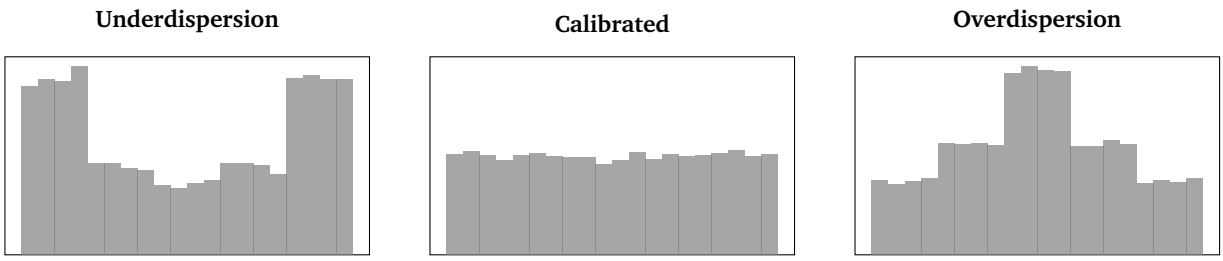


Figure 3.4.: A sketch illustrating how to interpret PIT/ verification rank histograms. The ensembles are underdispersed if there is more mass in the outer bins (left diagram) and overdispersed if there is more mass in the central bins (right diagram). Well-calibrated ensembles are uniformly distributed (middle diagram).

3.4 Evaluation

In this section, we evaluate the four post-processing strategies using the previously introduced setting of wind power forecasting. We start with the results for the benchmark data set (Section 3.4.1) before reporting the results for the Swedish data set (Section 3.4.2). This section only presents the results of different post-processing strategies using only the truncated normal distribution, with the results of further experiments with different distributions reported in Appendix A.

3.4.1 Benchmark Data

We first perform the different post-processing strategies on the benchmark data set and evaluate forecasting performance for both forecasting models with varying forecast horizons. Table 3.4 summarises the mean CRPS for the onshore and offshore benchmark data set with the best value highlighted in bold. We first observe that for every model and forecast horizon, at least one post-processing strategy performs better than the *Raw* strategy. Furthermore, for almost every model and every forecast horizon, either the *One Step-T* or *Two Step-WT* post-processing strategy results in the lowest CRPS. Which of these two post-processing strategies performs best depends both on the forecasting model used and the forecast horizon considered. For the offshore benchmark, for example, the *Two Step-WT* strategy is almost always better for the linear regression, whilst the *One Step-T* is always better for the random forest.

These results are further confirmed by Figure 3.5, which plots the CRPS skill score for each post-processing strategy across all forecast horizons. Considering the CRPS skill score, we can easily see that post-processing almost always leads to an improvement in CRPS when compared to the *Raw* strategy. The improvement depends on the forecast horizon, the forecasting model, and the data set but is typically between 5% and 15%. The noticeable exception is the *One Step-W* strategy, which, with the random forest on the offshore benchmarks, leads to worse CRPS performance for two of the considered forecast horizons.

Table 3.4.: Summary of mean CRPS on the test data for the benchmark data sets and for the linear regression (LR) and random forest (RF) on all forecast horizons. The best prediction for each strategy, forecast horizon, and model is highlighted in bold. [1]

Data Set		6h	12h	18h	24h
Onshore Benchmark	LR Raw	3.91	5.60	5.67	6.08
	LR *One Step-T*	3.53	**4.88**	5.71	**5.76**
	LR One Step-W	3.81	5.26	**5.51**	5.90
	LR *Two Step-WT*	**3.48**	4.97	5.68	5.77
	RF Raw	3.98	5.34	5.68	5.84
	RF *One Step-T*	**3.61**	4.63	**5.64**	**5.38**
	RF One Step-W	3.97	4.53	5.78	5.83
	RF *Two Step-WT*	3.67	**4.34**	**5.64**	5.70
Offshore Benchmark	LR Raw	18.98	23.51	25.46	24.11
	LR *One Step-T*	17.92	22.23	24.85	**23.44**
	LR One Step-W	18.74	23.31	24.51	23.88
	LR *Two Step-WT*	**17.30**	**22.08**	**24.28**	23.49
	RF Raw	20.77	23.80	25.00	25.54
	RF *One Step-T*	**19.47**	**22.15**	**24.24**	**23.83**
	RF One Step-W	21.45	23.22	25.41	24.93
	RF *Two Step-WT*	20.47	22.43	25.56	24.08

We next take a closer look at the calibration by considering PIT and verification rank histograms from our post-processing strategies. For brevity's sake, we focus on the onshore data set and pick a forecast horizon of 12h.[10]. First, we consider the uncalibrated weather EPSs in Figure 3.6. Each of these weather ensembles is underdispersed. The speed, temperature, and u10-wind rank histograms appear almost identical, with many observations in the lower left-hand bin, whilst the v10-wind also has many observations in the upper right-hand bin. Next, in Figure 3.7, we plot the PITs for the same weather ensembles after post-processing. We clearly observe an improvement in calibration. The speed, temperature, and u10-wind ensembles have almost perfectly uniform PITs. Only the v10-wind EPS demonstrates slight overdispersion.

Finally, we compare the calibration of the final wind power ensembles generated from the different post-processing strategies in Figure 3.8. The ensemble resulting from the *Raw* strategy is clearly underdispersed. Further, we observe that the ensemble from the *One Step-T* strategy is the most calibrated, with the associated PIT almost perfectly uniform. The calibration of the *One Step-W* strategy is also poor, with the resulting ensemble clearly underdispersed. Finally, the *Two Step-WT* strategy results in the second-best calibrated ensembles. Although not perfectly uniform, the resulting PIT is not obviously over- or underdispersed.

3.4.2 Swedish Data

For the Swedish data set, we also first consider the mean CRPS in Table 3.5 for each model and each forecast horizon. Again, we observe that the *Raw* strategy is never the best strategy,

[10]The other data set, forecasting model, and forecast horizons led to similar results and are, therefore, not presented in detail.

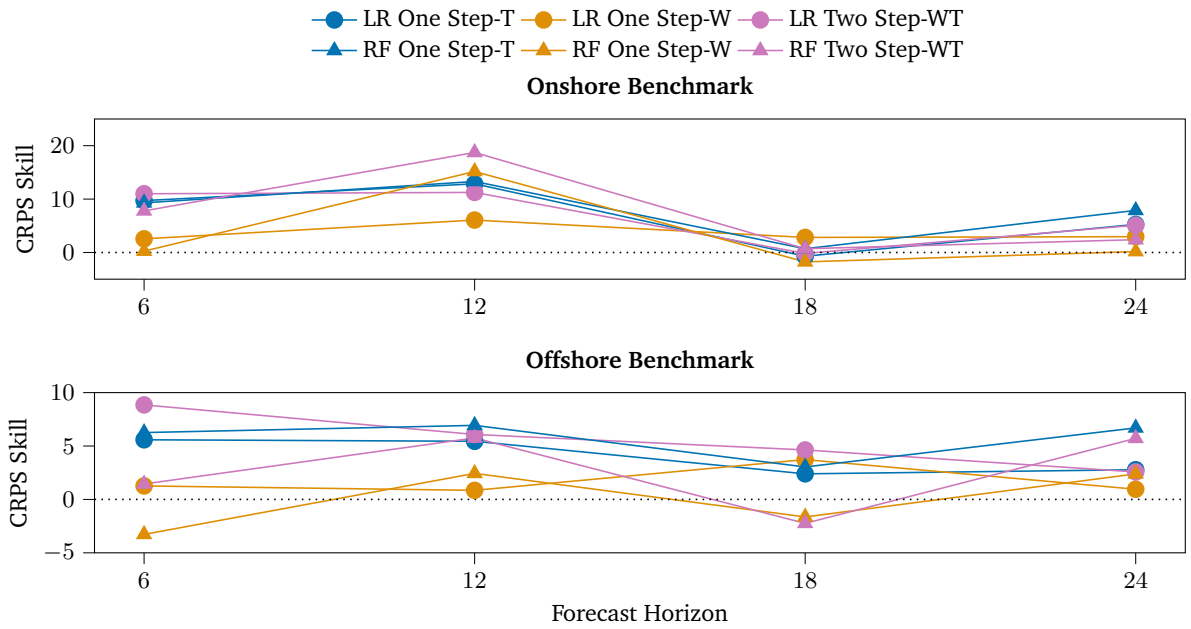
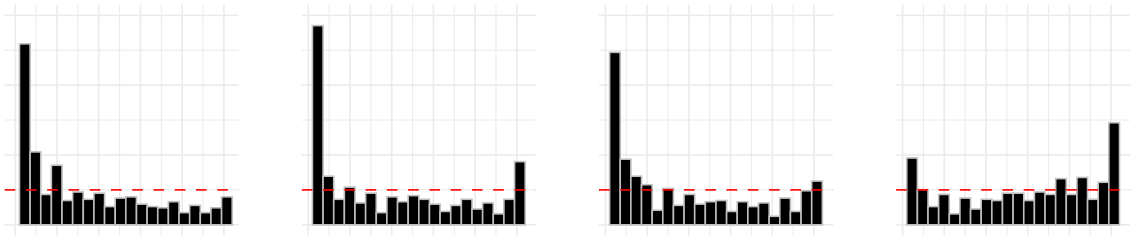


Figure 3.5.: The CRPS skill score plotted against the forecast horizon on the test data for the onshore benchmark (top figure) and the offshore benchmark (bottom figure). Positive values indicate an improvement over the *Raw* strategy in percent. In general, post-processing improves forecasting performance, although only post-processing the weather has little impact or performs slightly worse.

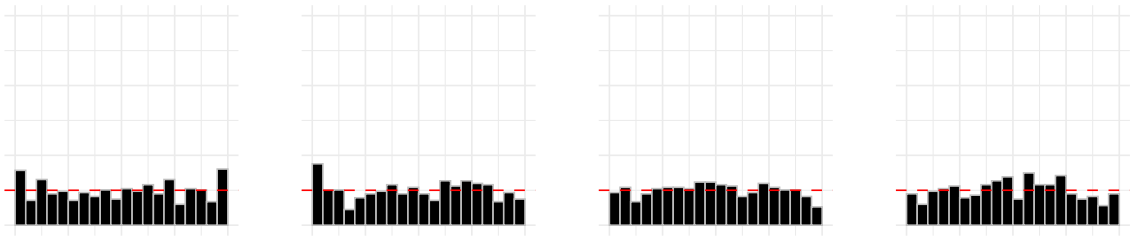


(a) Speed **(b)** Temperature **(c)** U10-Wind **(d)** V10-Wind

Figure 3.6.: An overview of the uncalibrated weather ensembles from the onshore data set used as inputs for our post-processing strategies. All weather ensembles are clearly underdispersed.

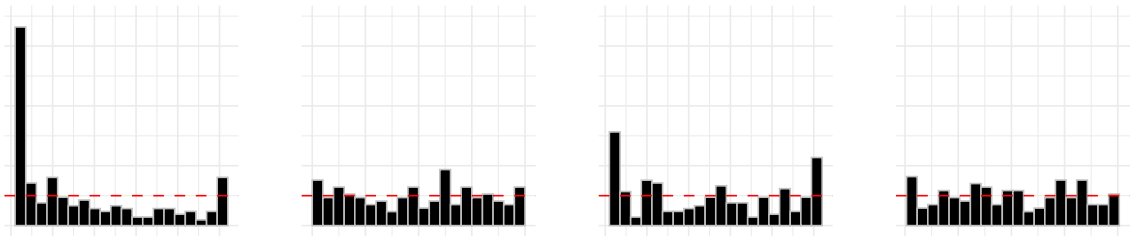
regardless of the forecasting model or horizon. Second, for both bidding zones in Sweden, the *One Step-W* is also never the best-performing post-processing strategy. Instead, for both forecasting models, data sets, and all forecasting horizons, either the *One Step-T* or *Two Step-WT* strategy performs best. In many cases, their performance is very similar or even identical.

These results are confirmed by considering the CRPS skill scores plotted in Figure 3.9. Again, we compare the improvement in CRPS with different post-processing strategies by using the *Raw* strategy as a baseline. These plots highlight several points. First, the *One Step-W* can sometimes be detrimental, leading to an increase in CRPS of up to 20% on the bidding zone 3 data set. Furthermore, on the bidding zone 3 data set, the post-processing strategies *One Step-T* and *Two Step-WT*, perform almost identically. Both strategies always lead to an improvement, and this improvement in CRPS can be as large as 40%, whilst not often dipping below 20%. Regarding bidding zone 4, the results are similar, if not as extreme. The *One Step-T* and *Two Step-WT*



(a) Speed **(b)** Temperature **(c)** U10-Wind **(d)** V10-Wind

Figure 3.7.: An overview of the calibrated weather ensembles from the onshore data set that are used as inputs into the forecasting models. All weather ensembles are far more calibrated than before.



(a) Raw **(b)** One Step-T **(c)** One Step-W **(d)** Two Step-WT

Figure 3.8.: An overview of the wind power ensembles resulting from the different post-processing strategies on the onshore data set. The *One Step-T* strategy results in the best-calibrated ensembles.

strategies again result in clear CRPS improvements, whilst only calibrating the weather ensembles in the *One Step-W* strategy leads to only slight improvements or slightly worse performance, depending on the forecast horizon. Finally, for both data sets, we notice that although the mean CRPS for different forecasting models (linear regression and random forests) are noticeably different, the skill scores are very similar. This can be observed for all three of the post-processing strategies.

3.5 Discussion

In this section, we briefly discuss the results of the evaluation of our post-processing strategies on the use case of wind power forecasting, present a few limitations of the considered use case, and mention some key insights.

Results The results of our evaluation show that post-processing always leads to more accurate wind power forecasts. However, the choice of strategy is important. It appears that the most important step is post-processing the final wind power ensembles since the two strategies which include post-processing of this ensemble (*One Step-T and Two Step-WT*) outperform the other strategies (*One Step-W* and *Raw*). This performance increase is noticeable in the lower CRPS

Table 3.5.: Summary of mean CRPS on the test data for the use case in Sweden for the linear regression (LR) and random forest (RF) on all forecast horizons. The best prediction for each strategy and each forecast model is highlighted in bold. [1]

Data Set		3h	6h	9h	12h	15h	18h	21h	24h
Bidding Zone 3	LR Raw	57.84	78.46	81.03	79.75	86.20	89.05	89.06	93.33
	LR *One Step-T*	47.30	55.99	**62.01**	67.39	67.31	64.36	67.17	68.00
	LR One Step-W	64.51	86.61	90.05	87.27	96.89	99.55	97.95	98.99
	LR *Two Step-WT*	**47.08**	**55.32**	62.04	**65.56**	**66.86**	**63.70**	**67.05**	**65.83**
	RF Raw	60.71	71.31	69.72	67.74	73.39	82.01	83.67	88.09
	RF *One Step-T*	46.08	50.28	**55.09**	55.96	**55.17**	**53.44**	57.83	58.86
	RF One Step-W	69.96	79.95	78.21	76.50	82.77	90.38	91.47	91.15
	RF *Two Step-WT*	**45.20**	**49.92**	55.48	**54.27**	55.36	53.95	**57.34**	**57.24**
Bidding Zone 4	LR Raw	41.75	49.91	50.93	63.17	63.85	53.27	51.46	51.21
	LR *One Step-T*	36.17	41.60	**43.15**	**49.68**	**53.51**	47.86	**45.92**	43.69
	LR One Step-W	41.91	51.20	51.85	57.17	61.52	52.86	51.24	48.86
	LR *Two Step-WT*	**35.78**	**41.24**	43.38	50.61	54.43	**47.50**	47.15	**43.14**
	RF Raw	36.97	41.33	40.74	50.67	51.60	45.54	42.26	39.41
	RF *One Step-T*	34.33	**38.25**	39.30	44.60	**48.02**	**44.28**	**41.72**	38.18
	RF One Step-W	36.81	42.91	41.94	46.51	51.16	46.24	43.16	38.31
	RF *Two Step-WT*	**33.52**	**38.25**	**39.21**	**43.95**	48.08	44.74	42.04	**37.33**

scores and in the better-calibrated wind power ensemble. Furthermore, only post-processing the weather (*One Step-W*) does not improve the forecast performance or ensemble calibration, and in some cases, it even leads to a decrease in performance. Importantly, this is despite the *One Step-W* leading to vastly improved calibration in the weather ensembles used as inputs to the forecasting model. A possible explanation for this behaviour is that post-processing only the weather neglects other bias sources for the wind power model. We train the models on historical data such that they learn the true relationship between the variables, however, this makes it harder for the forecasting model to properly propagate the uncertainty from the set of ensembles through the model. Thus, despite the weather ensembles being well-calibrated after post-processing, this information seems not to be accurately used by the forecasting model.

It is also worth noting that to enable comparisons across all strategies, we only consider a sample size of 51, i.e. the number of raw ensembles, when drawing from the calibrated weather ensembles. Although it is possible to vary the number of independently drawn samples, there are two reasons why we conclude this will not have a noticeable effect on the results. Firstly, the considered sample size of 51 is large enough to accurately approximate the estimated distribution and, secondly, increasing the number of samples will also increase computational complexity and could lead to poorer performance in subsequent calibration steps. However, a systematic analysis of the effect of sample size could be interesting in future work.

Limitations The first limitation of our evaluation is that we only consider wind power forecasts. Although we take different data sets and multiple forecasting models into account, the results of our post-processing strategy evaluation should be verified in other use cases where the target time series is affected by meteorological uncertainty. These time series could include solar power

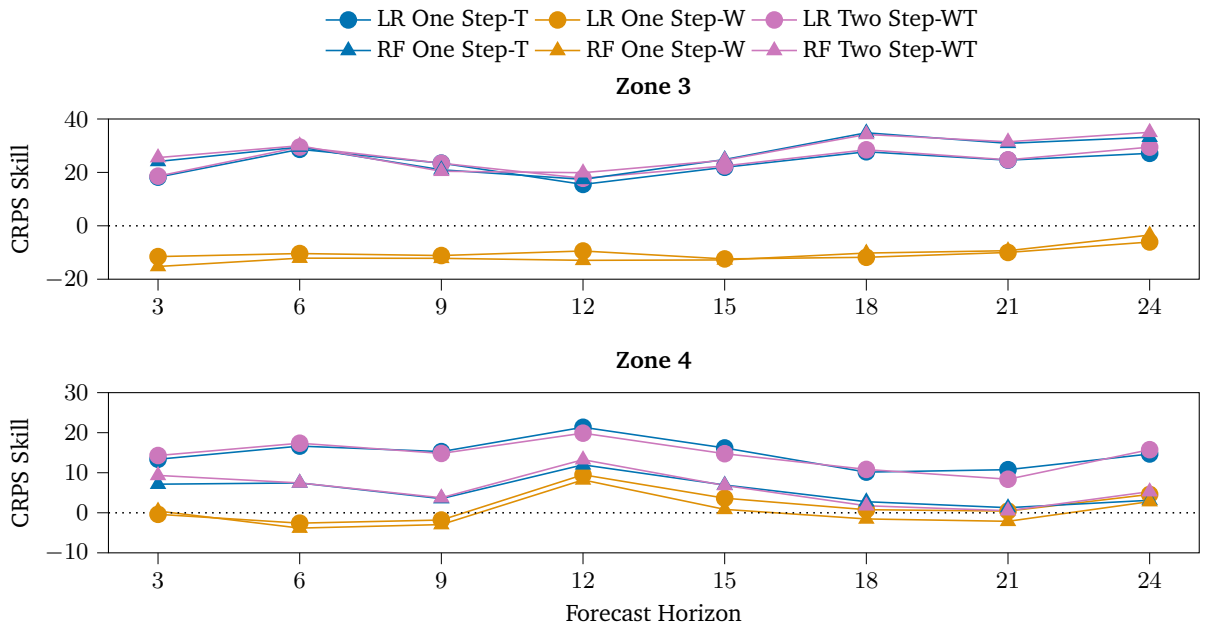


Figure 3.9.: The CRPS skill score plotted against the forecast horizon on the test data for both bidding zone 3 (top figure) and bidding zone 4 (bottom figure) in Sweden. Positive values indicate an improvement over the *Raw* strategy in percent. Post-processing the final power ensemble, either directly or as part of a two-step process, improves the forecast. However, only post-processing the weather ensemble leads to worse or similar forecast performance.

forecasting from photovoltaic or concentrating solar plants. Additionally, it may be interesting to consider certain sales time series since the meteorological connection in these time series is present but not as direct.

Second, we only consider EMOS as a post-processing method. Although EMOS is computationally cheap and effective, it assumes a parametric distribution, which may be limiting. Currently, modern machine learning approaches exist for ensemble post-processing, some of which are non-parametric and, therefore, do not make any distribution assumptions. Therefore, it would be wise to ratify our evaluation results with further non-parametric, post-processing methods.

Third, our evaluation only considered one-step-ahead forecasts for a single location or a broad region. Although we took different forecast horizons into account, we did not explicitly account for the temporal or spatial dependencies that occur for multi-step-ahead forecasts or when forecasting multiple similar locations simultaneously. It would be interesting to consider multiple-step-ahead forecasts either directly or by applying copulas to account for temporal dependencies similar to Grothe *et al.* [129].

Finally, our post-processing strategies are based on standard point forecasting models that do not learn the uncertainty in the EPS. Therefore, extending the evaluation to probabilistic models capable of considering probabilistic inputs and then learning the uncertainty themselves would be interesting.

Insights The first major insight from our evaluation is that post-processing is important. Furthermore, the post-processing strategy applied does affect the performance and calibration of the resulting forecast target ensemble and, therefore, should not be neglected.

From our proposed post-processing strategies, *One Step-T* and *Two Step-WT* both perform well, and in most cases, there is no noticeable difference between them. However, one factor differentiating the two strategies is the number of post-processing steps required. When we apply the *Two Step-WT* strategy, we need to post-process all weather inputs (in the case of the wind power use case, this is four, but in more complicated models, this number may increase). Once this post-processing is complete, we generate a probabilistic forecast based on these calibrated weather ensembles and are again required to process the resulting ensembles. The *One Step-T* strategy, on the other hand, involves only one post-processing step, irrespective of how many weather inputs the model includes. Although EMOS' computational complexity is low and multiple post-processing steps are still feasible, *One Step-T* is a more computationally efficient strategy that achieves similar forecasting accuracy. Therefore, our initial evaluation suggests that the last post-processing step has the largest impact on forecast performance and calibration and, due to its superior computational efficiency, should almost always be selected.

3.6 Conclusion

The present chapter aims to link the meteorological uncertainty explicitly contained in an Ensemble Prediction System (EPS) to target time series affected by this uncertainty. To achieve this connection, we evaluate four ensemble post-processing strategies to determine at which stage this post-processing is the most useful for probabilistic forecasts based on EPS input. More specifically, we identify four post-processing strategies that can be applied; (1) no post-processing (*Raw*), (2) a one-step strategy with post-processing of the resulting target forecast ensemble (*One Step-T*), (3) a one-step strategy with post-processing of the weather ensembles (*One Step-W*), and (4) a two-step strategy with both post-processing of the weather ensembles and the resulting target forecast ensembles (*Two Step-WT*). These strategies are evaluated using the Ensemble Model Output Statistics (EMOS) post-processing method and two different forecasting models, namely a linear regression and a random forest. We compare the results on four data sets, two synthetic benchmarks and two bidding zones from Sweden. Results show that post-processing generally improves performance, specifically when the wind power ensemble is post-processed. *One Step-T* and *Two Step-WT* deliver similar results in terms of Continuous Ranked Probability Score (CRPS) and CRPS Skill Score (CRPSS) performance, but since it requires significantly fewer post-processing steps, the *One Step-T* strategy is preferred.

Given these initial positive results, future work should extend the evaluation to other time series affected by meteorological uncertainty. Furthermore, our evaluation should be ratified by further ensemble post-processing methods, specifically, machine learning-based non-parametric methods. It may also be interesting to consider multi-step-ahead forecasts and multiple locations to include spatial and temporal dependencies in the evaluation. Ensemble Copula Coupling (ECC) or similar

may account for such dependencies. Finally, the current methods use traditional forecasting methods, which do not appear to be able to learn the uncertainty in the data. Therefore, approaches that learn this uncertainty in the weather ensembles and directly propagate it to a probabilistic forecast should be investigated.

4 Probabilistic Forecasts from the Underlying Data

The content of this chapter is based on:

K. Phipps *et al.*, "Generating probabilistic forecasts from arbitrary point forecasts using a conditional invertible neural network", *Applied Intelligence*, 2024. DOI: https://doi.org/10.1007/s10489-024-05346-9.

In the previous chapter, we quantified uncertainty by explicitly linking the meteorological uncertainty to the uncertainty in the target time series. However, this approach was limited to time series that are affected by meteorological conditions, and probabilistic forecasts are required to quantify the uncertainty associated with any prediction of the future [71], [130], not just those affected by the weather. Therefore, in the present chapter, we consider a general methodology for quantifying uncertainty that can also be applied to time series not dependent on meteorological conditions. Probabilistic forecasts of such time series are necessary for numerous applications, such as stabilising energy systems [131], managing congestion in traffic systems [132], or sizing servers of web applications to cope with peak web traffic [133]. However, despite this necessity for probabilistic forecasts, many modern forecasting methods still generate point forecasts [134]. Although many recent machine learning libraries offer support for probabilistic loss functions to simplify the generation of probabilistic forecasts, this does not take advantage of the existing point forecast model, which may be well-trained and designed for a specific application.

One solution to overcome this challenge is to generate probabilistic forecasts based on these existing point forecasts. For many years, such forecasts have been generated by analysing the residual errors of the point forecast and creating prediction intervals around the point forecast based on the standard deviation or quantiles of these errors [20], [135]. Moreover, such probabilistic forecasts can be generated by using machine learning methods exploiting the residual errors [136], [137], by applying the Bayesian theory of probability to a point method [138], or by considering Monte-Carlo sampling methods [139]. Although these methods may be effective, they also have various limitations. For example, the prediction interval-based approaches can only generate prediction intervals as probabilistic forecasts, while machine learning methods depend on the point forecast and must be retrained if the point forecast is altered. Ideally, such probabilistic forecasts should be generated directly from arbitrary point forecasts and should not require retraining if the point forecast changes.

Therefore, in the present chapter, we present a novel approach that generates probabilistic forecasts from arbitrary point forecasts by using a Conditional Invertible Neural Network (cINN)

to learn the underlying distribution of the time series data. Since time series have an inherent component of randomness [20], we aim to use this uncertainty within the distribution of the time series data to generate probabilistic forecasts. However, the underlying system responsible for this uncertainty generates observations of an unknown probability distribution. Therefore, with our approach, we first map this unknown probability distribution of the time series data to a known and tractable distribution by applying a cINN. Then, we use the output of a trained arbitrary point forecast method as an input to the trained cINN and consider the representation of this forecast in the known and tractable distribution. We then analyse the neighbourhood of this representation in the known and tractable distribution to quantify the uncertainty associated with the representation. Finally, we use the backward pass of the cINN to convert this uncertainty information into the forecast. In our approach, the cINN is trained independently of the point forecast and, therefore, must not be retrained when the point forecast is altered.

Thus, in the present chapter, we introduce our approach and empirically evaluate it using different data sets from various domains. The remainder of this chapter is structured as follows. First, we present related work and highlight the addressed research gap in Section 4.1. In Section 4.2, we then explain our approach in detail and highlight how we use a cINN to generate probabilistic forecasts from an arbitrary point forecast. We detail the experimental setup in Section 4.3, before presenting our results in Section 4.4. In Section 4.5, we discuss our evaluation and key insights. Finally, we conclude and suggest possible directions for future work in Section 4.6.

4.1 Related Work

The approach introduced in the present chapter is closely related to two fields: work that generates probabilistic forecasts based on point forecasts and work focusing on probabilistic forecasts using a cINN. In this section, we discuss related work from both fields.

Generating Probabilistic Forecasts from Point Forecasts Determining the uncertainty associated with a point prediction is one of the key research areas of uncertainty quantification [140]. Many methods focus on generating probabilistic prediction intervals from existing point forecasts by using the residual errors between the point forecast and the true value [20]. These prediction intervals can be generated by assuming a Gaussian distribution of the errors [20], using the empirical distribution of the errors [135], or considering nonconformity errors [141]–[143]. While effective, these methods are designed to generate prediction intervals rather than approximate the full probability distribution, which may be a limitation, particularly if these intervals are not generated from a parametric distribution.

Similar approaches also use residual errors in combination with further machine learning algorithms. Camporeale *et al.* [136], for example, train a neural network to forecast the standard deviation of the residual errors and generate probabilistic forecasts as realisations of a Gaussian distribution centred around the original point forecasts. Similarly, Wang *et al.* [137] use the

residual errors from a point forecast to train a Generative Adversarial Network (GAN). This trained GAN is then used to generate multiple residual scenarios, which are combined with the point forecast to form probabilistic forecasts. The main limitation of both approaches is that the additional machine learning models used to predict the uncertainty (i.e. standard deviation or residual scenarios) depend on the selected point forecast [136], [137]. Therefore, these machine learning models must be retrained whenever the point forecast is altered.

Further approaches include a Bayesian method involving assumed priors [138], [144], [145], integrating uncertainty into the prediction via an ensemble of predictions [74], [146], and considering uncertainty through Monte Carlo sampling approaches or similar [139], [147], [148]. The main limitation of these approaches, apart from the assumption regarding the Bayesian prior, is the computational complexity resulting from sampling or generating a large ensemble pool.

Probabilistic Forecasts using cINNs To generate probabilistic forecasts, cINNs, also referred to as normalising flows [149], are combined with other machine learning methods. Arpogaus *et al.* [150], for example, apply normalising flows to learn the parameters of Bernstein polynomials, which are in turn used to generate a probabilistic forecast. Moreover, Rasul *et al.* [151] combine normalising flows with recurrent neural networks to generate probabilistic forecasts. Normalising flows are also combined with quantile regression networks and copulas [152], or used to generate a conditional approximation of a Gaussian mixture model [153] to improve the accuracy of the resulting probabilistic forecasts. Whilst these methods are all effective, they use cINNs to enrich existing probabilistic forecasting methods but not to generate probabilistic forecasts.

An alternative method that directly uses normalising flows in the context of probabilistic forecasts is to learn multi-dimensional distributions of electricity price differences to predict the trajectory of intraday electricity prices [154]. Similarly, normalising flows may be applied multiple times to generate scenario-based probabilistic forecasts [155]–[157], or to generate a proxy for weather ensemble prediction systems based on numerical weather prediction models [158]. These methods use the generative nature of normalising flows to generate multiple predictions drawn from the same distribution. However, the forecasts are only probabilistic as an ensemble, with each individual forecast still being a point forecast. Furthermore, these forecasts rely on the assumption that the underlying learned distribution remains constant, and they do not always consider external features. Finally, these methods all focus on directly generating probabilistic forecasts and cannot be applied to generate probabilistic forecasts from existing point forecasts.

4.2 Generating Probabilistic Forecasts with a cINN

To generate probabilistic forecasts from arbitrary point forecasts, we directly apply the uncertainty in the underlying time series. This uncertainty usually reflects the inherent randomness or unpredictability of the measured underlying system. However, this underlying system typically generates observations of an unknown distribution. To solve this challenge, we aim to find a

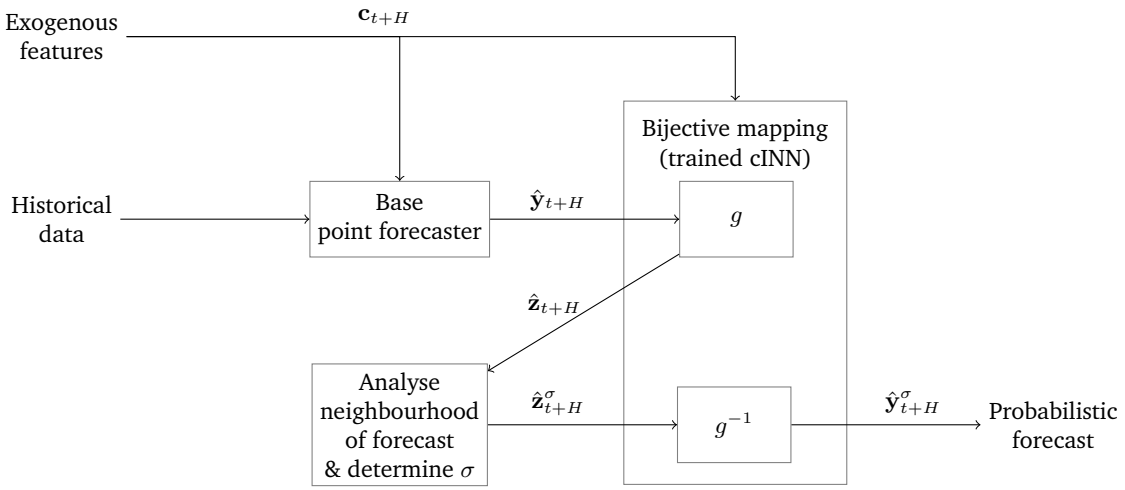


Figure 4.1.: Overview of the application of our approach. Exogenous features and historical data are used as inputs for an arbitrary base point forecaster. The resulting point forecast is combined with exogenous features as inputs to a bijective mapping realised by a trained cINN. This mapping generates a representation of the forecast in a known and tractable distribution. We analyse the neighbourhood of this known and tractable representation to include uncertainty. Finally, we map this representation back to the unknown distribution to generate a probabilistic forecast. Note that the training of the approach is not shown. [3]

bijective mapping from the unknown distribution to a known and tractable distribution. Since many time series are affected by exogenous features such as weather, this bijective mapping should also be able to consider such exogenous features, as shown in Figure 4.1. With the mapping g, we map a point forecast from the unknown distribution to its representation in a known and tractable distribution. In the known and tractable distribution, we analyse the neighbourhood of this representation and include uncertainty. Finally, we map this uncertainty information back to the unknown distribution using the inverse mapping g^{-1} to generate probabilistic forecasts.

4.2.1 Including Uncertainty from the Underlying Distribution of the Data

In this section, we first demonstrate that a bijective mapping from an unknown distribution in a known and tractable distribution exists. Given the existence of this mapping, we highlight the equivalence of the uncertainty in the image and the inverse image of the considered mapping. Finally, we describe how this mapping is realised with a cINN.

Bijective Mapping To introduce the bijective mapping, let us consider a times series $\mathbf{y} = \{y_t\}_{t \in T}$ consisting of T observations as realisations of a random variable $Y \sim f_Y(\mathbf{y})$ with a Probability Density Function (PDF) $f_Y(\mathbf{y})$ in the realisation space $\mathbb{Y}$. Furthermore, we have a bijective mapping $g : \mathbb{Y} \to \mathbb{Z}$ from the realisation space $\mathbb{Y}$ to the space of the tractable distribution $\mathbb{Z}$ where

$\mathbf{y} \mapsto g(\mathbf{y}, \circ) = \mathbf{z}$, and g being a continuously differentiable function.[1] To calculate the PDF $f_Z(\mathbf{z})$ in terms of $f_Y(\mathbf{y})$, we can apply the change of variables formula [159], [160], i.e.

$$f_Z(\mathbf{z}) = f_Y(g^{-1}(\mathbf{z}, \circ)) \left| \det \left(\frac{\partial g^{-1}}{\partial \mathbf{z}} \right) \right|, \tag{4.1}$$

where $\frac{\partial g^{-1}}{\partial \mathbf{z}}$ is the Jacobian matrix. Since g is bijective, this equation describes a bijective mapping from the unknown distribution $f_Y(\mathbf{y})$ to the known and tractable distribution $f_Z(\mathbf{z})$. Therefore, the change of variable formula provides us with the required mapping.

Equivalence of Uncertainty To show the equivalence of the uncertainty in the unknown distribution and known tractable distribution when applying Equation (4.1), we show the equivalence of quantiles in both distributions. To show this equivalence, we first consider the Cumulative Distribution Function (CDF) of the random variable $Z = g(Y, \circ) \sim f_Z(\mathbf{z})$, defined as

$$F_Z(\mathbf{z}) = \int_{-\infty}^{\mathbf{z}} f_Z(\mathbf{u}) d\mathbf{u}. \tag{4.2}$$

If we use the expression for $f_Z(\mathbf{z})$ from the change of variables formula (Equation (4.1)) in the definition of the CDF (Equation (4.2)), we obtain

$$F_Z(\mathbf{z}) = \int_{-\infty}^{\mathbf{z}} f_Y(g^{-1}(\mathbf{u}, \circ)) \left| \det \left(\frac{\partial g^{-1}}{\partial \mathbf{u}} \right) \right| d\mathbf{u}, \tag{4.3}$$

describing the CDF of $F_Z(\mathbf{z})$ in terms of the CDF $F_Y(\mathbf{y})$. Since g is a continuously differentiable function, we can apply integration by substitution to rewrite Equation (4.3) as

$$F_Z(\mathbf{z}) = \int_{-\infty}^{g^{-1}(\mathbf{z})} f_Y(\mathbf{v}) d\mathbf{v} = F_Y(g^{-1}(\mathbf{z}, \circ)), \tag{4.4}$$

which is simply the CDF of Y evaluated at the inverse of g. Further, the quantiles $\mathbf{z}^{(\alpha)}$ of Z are defined by the inverse of the CDF, i.e.

$$\begin{aligned} \mathbf{z}^{(\alpha)} = F_Z^{-1}(\alpha) &= \inf \mathbf{z} \mid F_Z(\mathbf{z}) \geq \alpha \\ &= \inf \mathbf{z} \mid F_Y(g^{-1}(\mathbf{z}, \circ)) \geq \alpha \end{aligned}$$

where $\inf$ refers to the infimum, the smallest value of $\mathbf{z}$ that fulfils the condition, and $\alpha \in (0, 1)$ is the considered quantile. Consequently, if we know that the α quantile of F_Z is $\mathbf{z}^{(\alpha)}$, then we can also calculate the α quantile of F_Y as $g^{-1}(\mathbf{z}^{(\alpha)}, \circ) = \mathbf{y}^{(\alpha)}$. From this follows an equivalence between the quantiles of Z and the quantiles of Y, which implies an equivalence in the uncertainty. Given the mathematical equivalence of the uncertainty in the two considered distributions, we can include uncertainty in a tractable and known distribution $f_Z(\mathbf{z})$ and use the inverse mapping $g^{-1} : \mathbb{Z} \to \mathbb{Y}$ to map this uncertainty to the original distribution $f_Y(\mathbf{y})$.

[1] The function g can include further parameters apart from $\mathbf{y}$, such as exogenous information. These further parameters are indicated via $\circ$.

Realising the Bijective Mapping To realise this bijective mapping g, we use a cINN [10], [149]. A cINN is a neural network that consists of multiple specially designed conditional affine coupling blocks [149]. As shown by Ardizzone *et al.* [149], these coupling blocks ensure that the mapping $g : \mathbb{Y} \rightarrow \mathbb{Z}$ learnt by the cINN is bijective. Furthermore, with the conditional information, the cINN is able to consider additional information, such as exogenous features, extracted statistical features from the time series, or calendar information, when learning the mapping [10]. As a result, the cINN is designed to learn an approximation of $f_Z(\mathbf{z})$ and a mapping g, which is per definition bijective, thus ensuring we can apply Equation (4.1) as described previously. Since cINNs are designed to efficiently calculate the inverse of a function [149], a well-trained cINN should be capable of learning the bijective mapping g, even if this mapping is non-trivial.

4.2.2 Applying our Approach

In the following, we describe how we realise the inclusion of uncertainty with a cINN.[2] We first detail how we train a cINN that learns the distribution of the underlying data. Second, we describe how we use this trained cINN to generate probabilistic forecasts.

Training We use a cINN to realise the continuous differentiable function g described above. The first input to this cINN is a segment $\mathbf{y}$ of the original time series, with the same length as the forecast horizon. More specifically, we train our cINN with multiple time series segments, all with the same length equal to the forecast horizon. In addition to this time series segment, we also consider conditional information $\mathbf{c}$ as an input to the function g. This conditional information always includes calendar features such as time of the day, and day of the week, but depending on the time series may also include additional exogenous features that are available for the forecast period. Thereby, the calendar information extracted from the time series is necessary conditional information to account for the temporal dependencies of the time series, whilst the exogenous features are optional. Furthermore, statistical features extracted from the time series can also be included as conditional information. The aim of the training is to ensure that the cINN learns the function g, so that resulting realisations $\mathbf{z} = g(\mathbf{y}, \mathbf{c})$ follow a known and tractable latent space distribution $f_Z(\mathbf{z})$. In our approach, we define this known and tractable latent space distribution as a multi-dimensional Gaussian distribution, where the number of dimensions is equal to the forecast horizon. Therefore, we apply the change of variables formula to derive the loss function

$$\mathcal{L}_{\text{cINN}} = \mathbb{E}\left[\frac{\parallel g(\mathbf{y}; \mathbf{c}, \theta) \parallel_2^2}{2} - \log \mid J \mid\right] + \lambda \parallel \theta \parallel_2^2, \tag{4.5}$$

where $J = \det(\partial g / \partial \mathbf{y})$ is the determinant of the Jacobian, θ is the set of all trainable parameters, and $\lambda \parallel \theta \parallel_2^2$ is an L2 regularisation [10], [149].[3] Training a cINN with this loss function results in a network with the optimised parameters $\hat{\theta}_{\text{OPT}}$ and ensures that the realised latent space

[2] The implementation to replicate the results of this chapter is available via GitHub: `https://github.com/KIT-IAI/ProbabilisticForecastsFromArbitraryPointForecasts`.

[3] Full details on the derivation of this loss function are presented in Ardizzone *et al.* [149].

distribution $f_Z(\mathbf{z})$ achieves the best possible approximation of the desired multi-dimensional Gaussian distribution [149]. Note that a base point forecaster must also be trained to apply our approach. However, since our approach enables probabilistic forecasts based on arbitrary point forecasts, this base point forecaster can be trained independently of the cINN, and the cINN must not be retrained if the point forecast is altered. Importantly, the output of the base forecaster must be a multi-horizon point forecast with the length of the forecast horizon, i.e. the length of the output from the base point forecaster must match the length of the time series segments used to train the cINN.

Forecasting To generate probabilistic forecasts, we begin with the H-step ahead point forecast $\hat{\mathbf{y}}_{t+H}$, which is the output of a base point forecaster. We combine this output with the associated conditional information for that H-step ahead forecast $\mathbf{c}_{t+H}$ and pass it on through the trained cINN to obtain a latent space representation of the output, i.e.

$$\hat{\mathbf{z}}_{t+H} = g(\hat{\mathbf{y}}_{t+H}, \mathbf{c}_{t+H}, \hat{\theta}_{\text{OPT}}). \tag{4.6}$$

Given this latent space representation of the point forecast, we explore the uncertainty in the neighbourhood of the forecast with

$$\tilde{\mathbf{z}}^i_{t+H} = \hat{\mathbf{z}}_{t+H} + \mathbf{r}^i, \quad i = 1, \dots, I, \ \mathbf{r}^i \sim \mathcal{N}(0, \sigma). \tag{4.7}$$

Using Equation (4.7), we select a random noise $\mathbf{r}^i$ from a standard normal distribution with mean 0 and variance σ and add this noise to the realisation $\hat{\mathbf{z}}$. We define the variance used for the sampling process σ as the *sampling hyperparameter*, which must be manually optimised using an evaluation metric. Due to the equivalence of uncertainty in both spaces shown in Section 4.2.1, we can process this perturbed sample via a backward pass of the cINN, i.e.

$$\tilde{\mathbf{y}}^i_{t+H} = g^{-1}(\tilde{\mathbf{z}}^i_{t+H}, \mathbf{c}_{t+H}, \hat{\theta}_{\text{OPT}}),$$

to obtain a perturbed sample in the realisation space $\tilde{\mathbf{y}}^i_{t+H}$. Based on the selected σ, we repeat the sampling process I times to obtain multiple realisations of $\tilde{\mathbf{z}}^i_{t+H}$ and, in turn, multiple realisations $\tilde{\mathbf{y}}^i_{t+H}$ that are all similar but not identical to the original forecast. If we combine all these samples in a set $\hat{\mathscr{Y}}^{\sigma}_{t+H}$, i.e.

$$\hat{\mathscr{Y}}^{\sigma}_{t+H} = \bigcup_{i \in I} \tilde{\mathbf{y}}^i_{t+H},$$

then this set of realisations provides a representation of the uncertainty in the neighbourhood of the forecast. Given this set, we calculate the quantiles which generates a probabilistic forecast derived from the original arbitrary point forecast, i.e. $\hat{\mathbf{y}}^{\sigma}_{t+H}$. It is important to note that the point forecast input to the cINN and the probabilistic forecast output are multi-step ahead forecasts with forecast horizon H. Therefore, the uncertainty is included in each multi-step forecast simultaneously, and we do not have to consider temporal correlations within each forecast explicitly.

4.3 Experimental Setup

This section describes the experimental setup we use to evaluate our approach. We first introduce the data used, before explaining the evaluation metrics. Furthermore, we describe the selected base forecasters used to generate the point forecasts, introduce the benchmarks we compare our approach to, and detail the implementation of the used cINN.

4.3.1 Data

We evaluate our proposed approach on four different data sets. The first data set is *Electricity*, namely the UCI Electricity Load Dataset[4] [161]. From this data set, we select the time series *MT_158* and resample it to an hourly resolution. The second data set, *Price*, contains zonal electricity price data recorded at a single location at an hourly resolution and taken from the electricity price track of the Global Energy Forecasting Competition 2014 (GEFCom2014) [162]. To evaluate our approach on a period longer than a single day, we combine data from all tasks in the GEFCom2014 price track. Third, we consider a *Solar* data set which contains hourly real-world solar power generation from a solar plant in Australia. This data set is taken from the solar power forecasting track of the GEFCom2014 [162] and, again, we combine data from all tasks to enable evaluation on a period longer than a day. The fourth data set, *Bike*, contains hourly records of rented bikes from the UCI Bikesharing Dataset [161], [163].[5] We normalise each of the above data sets before creating separate train, validation, and test subsets. An overview of these splits and the considered exogenous variables is presented in Table 4.1.

4.3.2 Evaluation Metrics

To evaluate our approach comprehensively, we consider the CRPS (see Equation (2.12)) as a measure for probabilistic forecast quality. Furthermore, to evaluate the calibration of our forecasts, we consider the Mean Absolute Quantile Deviation (MAQD) (see Equation (2.20)), and to measure sharpness, we evaluate the normalised Mean β-PI Width ($\mathrm{nMPI}(\beta)$) (see Equation (2.18)). Furthermore, as a measure for the quality of the prediction intervals, we consider the Mean Winkler (MW) score (see Equation (2.17)). In addition to the evaluation metrics for probabilistic forecasts, we also evaluate the quality of the base point forecasters. To this means we consider the Root Mean Squared Error (RMSE), which is given by

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{4.8}$$

with a true value y_i, a forecast value $\hat{y}_i$, and n observations.

[4] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
[5] https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset

Table 4.1.: Overview of the data sets used including the exogenous features considered, and the used train, validation, and test sets. [3]

Data Set	Target	Exogenous Features	Train	Validation	Test
Electricity	MT_158	Calendar Information[1]	$[0, 14716]$	$[14717, 21023]$	$[21024, 26280]$
Price	Zonal Price	Calendar Information[1], Forecast Total Load, Forecast Zonal Load	$[0, 14541]$	$[14542, 20773]$	$[20774, 25968]$
Solar	POWER	Calendar Information[1] SSRD[2], TCC[3]	$[0, 11033]$	$[11034, 15762]$	$[15763, 19704]$
Bike[4]	cnt	Calendar Information[1], Temperature, Humidity, Windspeed, Weather Situation	$[0, 9824]$	$[9825, 14034]$	$[14035, 17544]$

[1] Sine- and cosine-encoded time of the day, the sine- and cosine-encoded month of the year, and a Boolean that indicates whether the current day is a weekend day or not.
[2] Surface solar radiation downwards.
[3] Total cloud cover.
[4] To create a time index for this data, we merge the columns *dteday* and *hr* and deal with missing values using linear interpolation.

4.3.3 Selected Base Forecasters

We evaluate our approach on four simple and two state-of-the-art point forecasting methods. As simple base point forecasters we consider a *Linear Regression (LR)*, a *Random Forest (RF)*, a *Feed-Forward Neural Network (NN)*, and the *eXtreme Gradient Boosting (XGBoost)* Regressor. The two state-of-the-art base point forecasters are *Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS)* [164] and *Temporal Fusion Transformer (TFT)* [40]. We provide implementation details for each of the selected base point forecasters in Table 4.2. As inputs, all base forecasting methods receive 24 hours of historical information for the target time series and exogenous features for the forecast horizon. The exogenous features comprise calendar information and, depending on the data set, further exogenous variables, as shown in Table 4.1. When combining the base forecasters with the cINN to generate probabilistic forecasts, we manually select the σ that minimises the CRPS on the validation data set (see Table 4.3).

4.3.4 Probabilistic Benchmarks

To assess the quality of the probabilistic forecasts generated with our approach, we compare them to multiple probabilistic benchmarks. These benchmarks can be classified into the following two groups: probabilistic forecasts generated from existing point forecasts and directly generated probabilistic forecasts. In the following, we introduce the benchmarks of both groups.

Table 4.2.: Overview of the selected base forecasters used to generate point forecasts. [3]

Base Forecaster	Classification	Implementation Details	Library Used
LR	Statistical	Default Hyperparameters	SKlearn [165]
RF	Statistical	Default Hyperparameters	SKlearn [165]
NN	Machine Learning	**Hidden Layers:** 3 **Layer Sizes:** 90-64-32 **Hidden Activation Function:** relu **Output Activation Function:** linear **Optimiser:** Adam [166] **Batch Size:** 100 **Max Epochs:** 100	Tensorflow [167] Keras [168]
XGBoost	Gradient Boosting	Default Hyperparameters	XGBoost [169]
N-HiTS	Deep Learning	Default Hyperparameters	PyTorch Forecasting[3]
TFT	Deep Learning	Default Hyperparameters	PyTorch Forecasting[3]

[1] `https://pytorch-forecasting.readthedocs.io/en/stable/index.html`

Table 4.3.: The selected sampling hyperparameter for each base forecaster and each data set used in the evaluation. [3]

Data Set	LR	RF	NN	XGBoost	N-HiTS	TFT
Electricity	0.57	0.63	0.49	0.36	0.59	0.72
Price	0.73	0.98	0.92	0.48	0.69	0.76
Solar	0.14	0.77	0.21	0.45	0.22	0.44
Bike	0.54	0.97	0.57	0.46	0.33	0.36

Probabilistic Forecasts Based on Existing Point Forecasts The first group of probabilistic benchmarks considers methods that generate probabilistic forecasts from existing point forecasts. All of these benchmarks operate on a similar principle. They consider the empirical errors $\epsilon_i = \mid \hat{y}_i - y_i \mid$, between the point forecasts $\hat{y}_i$ and true values y_i on a validation data set. These empirical errors are then used to generate prediction intervals. The benchmarks differ in how these empirical errors are used to generate prediction intervals. The first benchmark is the *Gaussian Prediction Interval (Gaussian PI)*. In this case, the empirical errors are assumed to be distributed according to a Gaussian distribution and the prediction intervals are calculated based on the standard deviation of these errors [20]. Second, we consider the *Empirical Prediction Interval (Empirical PI)*. This benchmark does not assume any parametric distribution but instead uses the empirical distribution of these empirical errors to calculate the prediction intervals [135]. Finally, we consider a *Conformal Prediction Interval (Conformal PI)*. This benchmark, introduced for multi-horizon time series forecasts by [141], calculates a critical nonconformity score for each of the empirical errors and applies Bonferroni and finite sample correction to ensure temporal dependence across the forecast horizon. These nonconformity scores are combined with the point forecast to generate the prediction intervals [141].

Table 4.4.: The architecture of the used cINN. [3]

Parameter	Description
Layers per block	Glow coupling layer and random permutation
Subnetwork in block	Fully connected (see Table 4.5)
Number of blocks	5
Conditioning network	Fully connected (see Table 4.5)

Direct Probabilistic Forecasts The second group of probabilistic benchmarks considers methods that directly generate probabilistic forecasts. The first of these benchmarks is *DeepAR* [170], which is an autoregressive recurrent neural network-based approach for probabilistic forecasting. We implement DeepAR using the PyTorch Forecasting library[6]. The second benchmark method is a *Quantile Regression Neural Network (QRNN)*. It trains a NN to directly forecast selected or multiple quantiles instead of the mean or median [171]. To realise the QRNN, we use a separate simple feed-forward NN to forecast each of the selected quantiles training each NN with the appropriate pinball loss function. The QRNN is implemented using TensorFlow [167] with the Keras [168] library and the pinball loss function. The third benchmark method uses the *Nearest Neighbour Quantile Filter (NNQF)* [172]. Similar to the QRNN, this method also forecasts quantiles. However, instead of using a custom quantile loss function to directly learn the quantiles, the NNQF finds similar values for each time step based on similarity in the target variable to determine quantiles in the data. A forecasting method is then trained to predict these calculated quantiles [172]. To realise the NNQF, we use a multi-layer feed-forward NN with one output per quantile, which is implemented using sklearn [165] and pyWATTS [7].

4.3.5 Used cINN

In the evaluation, we use the same cINN architecture (see Table 4.4) for each of the considered data sets. It is based on Generative Flow with Invertible 1×1 Convolutions (GLOW) coupling layers that consider conditional input [173]. These GLOW layers consisted of an activation normalisation layer, a 1×1 convolution layer, and an affine coupling layer first introduced by Dinh *et al.* [174]. Each of these layers is designed to be reversible so that the resulting GLOW layer is also reversible and the cINN can be trained in both directions [166], [174]. Similar to Heidrich *et al.* [10], the conditional input is provided by a fully connected NN, which uses the same exogenous information available to the base forecaster as conditional information (see Table 4.1). We detail the implementation information for the used cINN in Table 4.4 and Table 4.5. When training the used cINN, we apply the Adam optimiser with a maximum of 100 epochs. Furthermore, when sampling in the latent space to generate probabilistic forecasts, we consider a sample size of 100. We implement the cINN in a pipeline with pyWATTS [7].

[6] `https://pytorch-forecasting.readthedocs.io/en/stable/api/pytorch_forecasting.models.deepar.DeepAR.html`

Table 4.5.: Implementation details of the subnetwork and the conditioning network in the used cINN. [3]

(a) Subnetwork.

Layer	Description
Input	[Output of previous coupling layer, conditional information]
1	Dense 32 neurons; activation: tanh
2	Dense 24 neurons; activation: linear

(b) Conditioning network.

Layer	Description
Input	[Calendar information, historical information, exogenous forecasts if available]
1	Dense 8 neurons; activation: tanh
2	Dense 4 neurons; activation: linear

Table 4.6.: Comparison of the average RMSE on the test data set for each of the considered point forecasters. The best values for each data set are highlighted in bold. [3]

Data Set	Electricity	Price	Solar	Bike
LR	0.5246	0.4118	0.3331	0.8565
RF	0.4601	0.4253	**0.2891**	0.9913
NN	0.4894	0.4499	0.3257	0.9227
XGBoost	**0.4532**	0.4090	0.2966	0.9258
N-HiTS	0.5329	0.4124	0.3686	0.6471
TFT	0.5134	**0.3672**	0.3366	**0.5457**

4.4 Evaluation

We evaluate our proposed approach in two steps. First, we compare the probabilistic forecasts generated from our approach when using different base point forecasters. Second, we compare our approach with existing probabilistic benchmarks. Throughout the evaluation for all data sets, we only consider a forecast horizon of $24\,\mathrm{h}$ and always generate multi-step ahead forecasts.

4.4.1 Comparison of Different Base Point Forecasters

In this section, we compare the performance of the different base point forecasts in our approach. First, we analyse the stand-alone performance of the point forecasters by reporting the average RMSE (Equation (4.8)) over five runs in Table 4.6. In general, we observe that the best-performing point forecast depends on the data set considered, with the performance of most point forecasters varying noticeably across the data sets. However, the TFT performs most consistently, achieving the lowest RMSE on two of the four data sets. In the remainder of this section we evaluate the probabilistic forecasts generated when combining these point forecasts with the cINN by comparing the forecast quality, the calibration, the sharpness, and the prediction intervals.

Table 4.7.: The average CRPS calculated on the test data set for each of the considered base point forecasters combined with the cINN over five runs. The best values for each data set are highlighted in bold. [3]

Data Set	Electricity	Price	Solar	Bike
LR-cINN	0.3180	0.1641	0.1686	0.4481
RF-cINN	0.2339	0.1689	**0.1056**	0.4992
NN-cINN	0.2542	0.1721	0.1399	0.4493
XGBoost-cINN	**0.2337**	0.1565	0.1072	0.4561
N-HiTS-cINN	0.2844	0.1552	0.1705	0.3454
TFT-cINN	0.2588	**0.1404**	0.1233	**0.2641**

Table 4.8.: The average MAQD between the theoretical and forecast quantiles calculated on the test data set for each of the considered base point forecasters combined with the cINN over five runs. The best values for each data set are highlighted in bold. [3]

Data Set	Electricity	Price	Solar	Bike
LR-cINN	0.1360	**0.0721**	0.2079	0.1324
RF-cINN	0.1009	0.1246	0.0666	0.2023
NN-cINN	0.0886	0.1136	0.1780	0.1605
XGBoost-cINN	**0.0811**	0.0948	**0.0431**	0.1369
N-HiTS-cINN	0.0929	0.1010	0.1953	0.1236
TFT-cINN	0.0817	0.0959	0.1420	**0.0945**

Quality For each of the base point forecasters combined with the cINN, we report the average CRPS across five runs in Table 4.7. We observe that the best-performing point forecaster combined with the cINN depends on the data set considered, although the TFT base point forecaster combined with the cINN results in the lowest CRPS on two of the four data sets. Furthermore, we observe that all base point forecasters combined with the cINN perform similarly on the Electricity, Price, and Solar data set, with the difference between the best and worst performing base point forecaster never larger than 0.0801. However, for the bike data set the difference between the TFT as the best performing base point forecaster when combined with the cINN, and the RF as the worst performing base point forecaster is 0.2351. Therefore, not only the absolute result but also the variance between the results can be data set dependent.

Calibration To analyse the calibration of the probabilistic forecasts generated by combining different point forecasters with the cINN, we report the MAQD in Table 4.8. We observe that the calibration performance again depends on the considered data set. The cINN combined with XGBoost as a base point forecaster results in the lowest MAQD on two of the four data sets, whilst using a LR as the base point forecaster results in the lowest MAQD on the Price data set, and the TFT when combined with the cINN performs best on the Bike data set. Furthermore, the MAQD varies noticeably between the data sets. On the Electricity and Price data sets, the difference in MAQD between the best and worst base point forecaster when combined with the cINN is always under 0.0549. However, on the Bike data set, this difference in MAQD is 0.1078, whilst on the Solar data set the difference between the best and worst performing base point forecaster when combined with the cINN is even larger at 0.1648.

Table 4.9.: The average $\mathrm{nMPI}(\beta)$ calculated on the test data set for each of the considered base point forecasters combined with the cINN for $1-\beta = 98\%$ and $1-\beta = 70\%$ over five runs. The best values for each data set and β are highlighted in bold. [3]

Data Set	Electricity		Price		Solar		Bike	
	98%	70%	98%	70%	98%	70%	98%	70%
LR-cINN	1.2953	0.5653	0.4618	0.2043	0.8989	0.4146	1.1744	0.5085
RF-cINN	**0.7795**	**0.3640**	0.4619	0.2046	**0.6268**	**0.2964**	1.1179	0.4815
NN-cINN	1.0163	0.4571	0.5785	0.2530	0.8319	0.3821	1.1661	0.5014
XGBoost-cINN	0.8248	0.3803	0.4518	0.2022	0.7510	0.3362	1.1335	0.4915
N-HiTS-cINN	1.0811	0.4837	0.3741	0.1690	0.9315	0.4061	**0.6906**	**0.3160**
TFT-cINN	1.1376	0.5145	**0.2893**	**0.1329**	0.7408	0.3462	0.7051	0.3252

Table 4.10.: The average MW score calculated on the test data set for each of the considered base point forecasters combined with the cINN over five runs. The best values for each data set are highlighted in bold. [3]

Data Set	Electricity	Price	Solar	Bike
LR-cINN	30.0248	14.6476	10.7391	36.8567
RF-cINN	**17.2494**	13.9050	**6.7742**	35.9929
NN-cINN	21.5586	16.4061	9.3110	35.4235
XGBoost-cINN	19.4470	13.6459	7.3523	35.4601
N-HiTS-cINN	25.7098	12.5044	11.1579	27.2643
TFT-cINN	22.1819	**11.1381**	7.9020	**22.0296**

Sharpness We evaluate the sharpness of the probabilistic forecasts generated when combining different base learners with the cINN by reporting the average $\mathrm{nMPI}(\beta)$ over five runs in Table 4.9. As with the calibration results, we observe best-performing base point forecasters when combined with the cINN depends on the data set. The RF, when combined with the cINN, achieves the best $\mathrm{nMPI}(\beta)$ on two of the four data sets, whilst the TFT as the base point forecaster performs best on the Price data set, and N-HiTS as the base point forecaster performs best on the Bike data set. We again observe varying $\mathrm{nMPI}(\beta)$s across the data sets, with the largest $\mathrm{nMPI}(\beta)$ of 1.2953 for $1-\beta = 98\%$ on the Electricity data set and the narrowest $\mathrm{nMPI}(\beta)$ of 0.1329 for $1-\beta = 70\%$ on the Price data set.

Prediction Intervals We report the average MW score across five runs as a measure of the quality of the prediction intervals generated by different point forecasters in Table 4.10. The performance varies depending on the considered data set. Probabilistic forecasts generated when combining the TFT with the cINN result in the lowest MW score for the Price and Bike data sets, whilst the RF as the base point forecaster results in the lowest MW score for the Electricity and Solar data sets. Regarding the MW scores, the performance varies noticeably depending on which base point forecaster is used on all data sets except for the Price data set, where the performance is similar for all base point forecasters when combined with the cINN.

Table 4.11.: A comparison of the average CRPS when generating probabilistic forecasts based on existing point forecasts. The average CRPS is calculated across five runs on the test data set, and the best values for each base point forecaster in each data set are highlighted in bold. [3]

Data	Point Forecaster	cINN	Gaussian PI	Empirical PI	Conformal PI
Electricity	XGBoost	**0.2337**	0.2993	0.2341	0.2339
	N-HiTS	0.2844	0.3630	0.2840	**0.2835**
	TFT	**0.2588**	0.3543	0.2658	0.2657
Price	XGBoost	**0.1565**	0.3496	0.1789	0.1786
	N-HiTS	**0.1552**	0.2785	0.1713	0.1712
	TFT	**0.1404**	0.2959	0.1607	0.1608
Solar	XGBoost	**0.1072**	0.2083	0.1249	0.1250
	N-HiTS	**0.1705**	0.2713	0.1781	0.1777
	TFT	**0.1233**	0.2333	0.1398	0.1398
Bike	XGBoost	**0.4561**	0.6075	0.4857	0.4856
	N-HiTS	0.3454	0.4232	0.3368	**0.3363**
	TFT	**0.2641**	0.3597	0.2680	0.2679

4.4.2 Comparison to Benchmarks

In the second step of our evaluation, we compare probabilistic benchmarks with the probabilistic forecasts generated when combining a cINN with the XGBoost, N-HiTS, and TFT base point forecasters. First, we compare the probabilistic forecasts from our approach to benchmarks that also use these same point forecasters to generate probabilistic forecasts. Second, we compare our approach to methods that directly generate probabilistic forecasts.

Probabilistic Forecasts Based on Existing Point Forecasts

In this section, we again analyse quality, calibration, sharpness, and the prediction intervals for each considered data set.

Quality We evaluate the quality of the different probabilistic forecasts by reporting the average CRPS across five runs in Table 4.11. We first observe that our approach using a cINN always performs better or similarly to the benchmarks. The cINN results in probabilistic forecasts with the lowest CRPS in all cases, except for N-HiTS when combined with the cINN on the Electricity and Bike data sets. In these two cases, the Conformal PI results in the lowest CRPS, however, the CRPS resulting from our approach is only 0.009 larger on the Electricity data set and 0.0091 larger on the Bike data set. Second, we observe that the Gaussian PI consistently results in the highest CRPS. Finally, we note that, across all data sets and for all considered point forecasters, the Empirical PI generates probabilistic forecasts resulting in almost identical CRPSs to those from the Conformal PI.

Table 4.12.: Comparison of the average MAQD when generating probabilistic forecasts based on existing point forecasts. The average MAQD is calculated using five runs on the test data set, and the best values for each base point forecaster in each data set are highlighted in bold. [3]

Data	Point Forecaster	cINN	Gaussian PI	Empirical PI	Conformal PI
Electricity	XGBoost	0.0811	0.1029	0.0221	**0.0220**
	N-HiTS	0.0929	0.1034	0.0117	**0.0112**
	TFT	0.0817	0.1088	0.0241	**0.0239**
Price	XGBoost	0.0948	0.1565	0.0390	**0.0387**
	N-HiTS	0.1010	0.1467	0.0251	**0.0245**
	TFT	0.0959	0.1534	**0.0349**	0.0351
Solar	XGBoost	0.0431	0.1252	**0.0124**	0.0125
	N-HiTS	0.1953	0.1239	**0.0186**	0.0194
	TFT	0.1420	0.1290	**0.0215**	**0.0215**
Bike	XGBoost	0.1369	**0.1169**	0.1273	0.1271
	N-HiTS	0.1236	0.1099	0.0286	**0.0281**
	TFT	0.0945	0.1169	**0.0110**	0.0111

Calibration To evaluate the calibration of the considered probabilistic forecasts, we report the average MAQD for each data set calculated across five runs in Table 4.12. We first observe that the results depend strongly on the base point forecaster and the data set considered. Whilst the Conformal PI results in the lowest MAQD for all base point forecasters on the Electricity data set, the results for the other data sets are not as clear. On the Price and Solar data sets, the Conformal PI or Empirical PI achieve the lowest MAQD depending on the considered point forecaster, whilst either Conformal PI, Empirical PI or Gaussian PI perform best on the Bike data set. Our approach using the cINN never achieves the lowest MAQD.

Sharpness To assess the sharpness of probabilistic forecasts generated from point forecasts, we report the average $\mathrm{nMPI}(\beta)$ over five runs in Table 4.13. Our approach using a cINN results in the lowest $\mathrm{nMPI}(\beta)$ for all base point forecasters, considered values of β and data sets in all but three cases. These exceptions are when the TFT is used as a base point forecaster for $1-\beta = 70\%$ on the Electricity and Solar data sets, and XGBoost as the base point forecaster for $1-\beta = 70\%$ on the Solar data set. Moreover, the $\mathrm{nMPI}(\beta)$s from the Empirical PI and Conformal PI are generally the largest for $1-\beta = 98\%$, and noticeably so. For example, the $\mathrm{nMPI}(\beta)$s for $1-\beta = 98\%$ for Conformal PI on the Electricity, Price, and Solar data sets are at least double the $\mathrm{nMPI}(\beta)$s generated with the cINN, and still noticeably larger on the Bike data set. Further, the $\mathrm{nMPI}(\beta)$s for $1-\beta = 70\%$ are generally the largest with the Gaussian PI. In general, the $\mathrm{nMPI}(\beta)$s vary noticeably depending on the data set and selected point forecaster.

Prediction Intervals To simultaneously consider calibration and sharpness, we analyse the prediction intervals of the considered probabilistic forecasts by comparing the average MW scores for each data set calculated over five runs in Table 4.14. We note that the probabilistic forecasts generated with the cINN result in the lowest MW scores on all data sets and for all considered point forecasters. Furthermore, the Winkler scores from our approach are noticeably smaller than the benchmarks. Although all the prediction interval-based benchmarks generate probabilistic

Table 4.13.: Comparison of the average $\mathrm{nMPI}(\beta)$ when generating probabilistic forecasts based on existing point forecasts for $1-\beta = 98\%$ and $1-\beta = 70\%$. The average $\mathrm{nMPI}(\beta)$ is calculated using five runs on the test data set, and the best values for each base point forecaster and β in each data set are highlighted in bold. [3]

Data	Point Forecaster	PI	cINN	Gaussian PI	Empirical PI	Conformal PI
Electricity	XGBoost	98%	**0.8248**	1.5418	1.8594	1.8527
		70%	**0.3803**	0.9276	0.4510	0.4529
	N-HiTS	98%	**1.0811**	1.8780	2.1627	2.1579
		70%	**0.4837**	1.1278	0.6265	0.6265
	TFT	98%	**1.1376**	1.8769	2.2586	2.2469
		70%	0.5145	1.1288	**0.4886**	0.4965
Price	XGBoost	98%	**0.4518**	1.5271	1.9885	1.9193
		70%	**0.2022**	0.8810	0.2434	0.2444
	N-HiTS	98%	**0.3741**	1.1536	1.4098	1.3963
		70%	**0.1690**	0.6493	0.2352	0.2364
	TFT	98%	**0.2893**	1.3332	1.7665	1.7534
		70%	**0.1329**	0.7571	0.2589	0.2592
Solar	XGBoost	98%	**0.7510**	1.8661	2.4523	2.4564
		70%	0.3362	1.0729	**0.2617**	0.2626
	N-HiTS	98%	**0.9315**	2.3938	2.9698	2.9618
		70%	**0.4061**	1.3934	0.5539	0.5475
	TFT	98%	**0.7408**	2.0298	2.6828	2.6830
		70%	0.3462	1.1668	**0.2651**	0.2668
Bike	XGBoost	98%	**1.1335**	1.8288	1.9887	1.9885
		70%	**0.4915**	1.1201	0.6329	0.6342
	N-HiTS	98%	**0.6906**	1.3939	1.5342	1.5299
		70%	**0.3160**	0.8199	0.4735	0.4749
	TFT	98%	**0.7051**	1.2546	1.4288	1.4344
		70%	**0.3252**	0.7332	0.3695	0.3706

forecasts with similar Winkler scores, the Gaussian PI results in slightly lower Winkler scores on all data sets. Finally, we observe that similar to the CRPS results, the MW scores for the Empirical PI and Conformal PI are almost identical for every data set and each considered point forecaster.

Direct Probabilistic Forecasts

To evaluate the performance of our approach when compared to benchmarks that directly generate probabilistic forecasts, we again consider the forecast quality, calibration, sharpness and the prediction intervals for each of the four considered data sets.

Quality To analyse the quality of the probabilistic forecasts, we report the average CRPS across five runs for all data sets in Table 4.15. The first observation is that our approach results in the lowest CRPS on three of the four data sets. Thereby, the choice of the base point forecaster is important, with XGBoost combined with the cINN performing best on the Electricity data set, whilst the TFT combined with the cINN performs best on the Price and Bike data sets. On the Solar data set the QRNN benchmark model outperforms all others, although our approach using XGBoost as a base point forecaster performs similarly, with a difference of only 0.0059. In general,

Table 4.14.: Comparison of the average MW scores when generating probabilistic forecasts based on existing point forecasts. The average MW score is calculated across five runs on the test data set, and the best values for each base point forecaster in each data set are highlighted in bold. [3]

Data	Point Forecaster	cINN	Gaussian PI	Empirical PI	Conformal PI
Electricity	XGBoost	**19.4470**	35.5141	39.5519	39.4368
	N-HiTS	**25.7098**	44.5813	47.7180	47.5730
	TFT	**22.1819**	43.4597	48.7009	48.5111
Price	XGBoost	**13.6459**	47.6781	56.1564	54.8959
	N-HiTS	**12.5044**	35.9409	39.0527	39.0557
	TFT	**11.1381**	40.1158	46.8259	46.8474
Solar	XGBoost	**7.3523**	25.6594	29.7955	29.8464
	N-HiTS	**11.1579**	34.2878	38.5077	38.4250
	TFT	**7.9020**	28.6166	32.9194	32.9338
Bike	XGBoost	**35.4601**	70.4294	72.7355	72.7596
	N-HiTS	**27.2643**	51.1508	52.0275	51.8381
	TFT	**22.0296**	44.1040	46.3018	46.3026

Table 4.15.: Comparison of the average CRPS between the probabilistic forecasts from the cINN and the direct probabilistic benchmarks. The average CRPS is calculated over five runs on the test data set, and the best values for each data set are highlighted in bold. [3]

Data Set	Electricity	Price	Solar	Bike
XGBoost-cINN	**0.2337**	0.1565	0.1072	0.4561
N-HiTS-cINN	0.2844	0.1552	0.1705	0.3454
TFT-cINN	0.2588	**0.1404**	0.1233	**0.2641**
DeepAR	0.3115	0.1583	0.1509	0.2985
QRNN	0.2866	0.1571	**0.1013**	0.4431
NNQF	0.2629	0.1825	0.1191	0.5415

the performance of the direct benchmarks is also highly dependent on the considered data set. Of the direct benchmarks, the NNQF performs best for the Electricity data set, the QRNN for the Price and Solar data sets, and DeepAR for the Bike data set.

Calibration To assess the calibration, we report the average MAQD across five runs in Table 4.16. Similar to the CRPS results, our approach using a cINN results in the lowest deviation for three of the four data sets. The lowest MAQD is achieved when combining the cINN with XGBoost as the base point forecaster for the Electricity and Solar data sets, and with the TFT on the Bike data set. On the Price data set, the NNQF achieves the lowest overall MAQD, outperforming all other benchmarks, however our approach combining XGBoost with the cINN achieves the second best MAQD. With regards to the direct benchmarks, the QRNN performs best on the Solar data set whilst the NNQF has the best performance of the direct benchmarks regarding the MAQD on all other data sets.

Sharpness To compare the sharpness of probabilistic forecasts, we report the average $\mathrm{nMPI}(\beta)$ over five runs in Table 4.17. With regards to the $\mathrm{nMPI}(\beta)$, our approach results in the smallest

Table 4.16.: Comparison of the average MAQD between the theoretical and forecast quantiles from the cINN and the direct probabilistic benchmarks. The average MAQD is calculated across five runs on the test data set, and the best values for each data set are highlighted in bold. [3]

Data Set	Electricity	Price	Solar	Bike
cINN-XGBoost	**0.0811**	0.0948	**0.0431**	0.1369
cINN-N-HiTS	0.0929	0.1010	0.1953	0.1236
cINN-TFT	0.0817	0.0959	0.1420	**0.0945**
DeepAR	0.2222	0.2139	0.2301	0.2478
QRNN	0.1420	0.1332	0.0708	0.2110
NNQF	0.0835	**0.0692**	0.1282	0.1303

Table 4.17.: Comparison of the average $\mathrm{nMPI}(\beta)$ between forecasts from the cINN and the direct probabilistic benchmarks for $1-\beta = 98\%$ and $1-\beta = 70\%$. The $\mathrm{nMPI}(\beta)$ is calculated across five runs on the test data set, and the best values for each data set and β are highlighted in bold. [3]

Data Set	Electricity		Price		Solar		Bike	
	98%	70%	98%	70%	98%	70%	98%	70%
XGBoost-cINN	**0.8248**	**0.3803**	0.4518	0.2022	0.7510	**0.3362**	1.1335	0.4915
N-HiTS-cINN	1.0811	0.4837	0.3741	0.1690	0.9315	0.4061	**0.6906**	**0.3160**
TFT-cINN	1.1376	0.5145	**0.2893**	**0.1329**	**0.7408**	0.3462	0.7051	0.3252
DeepAR	1.1896	0.5204	0.5850	0.2672	1.9875	0.8517	1.1524	0.5204
QRNN	1.6227	0.5990	0.6956	0.2840	1.5715	0.6610	1.6535	0.4150
NNQF	1.4587	0.7790	0.8259	0.3708	1.8035	0.8962	1.5027	0.7468

$\mathrm{nMPI}(\beta)$ for all data sets. Using the TFT as a base point forecaster generates the narrowest prediction intervals for the Price data set, and for $1-\beta = 98\%$ on the Solar data set. Combining XGBoost as a base point forecaster with the cINN results in the narrowest prediction intervals for the Electricity data set and $1-\beta = 70\%$ on the Solar data set, whilst N-HiTS combined with the cINN generates the narrowest prediction intervals on the Bike data set. The width of the prediction intervals for the direct probabilistic benchmark depends on the data set. For the Electricity and Price data sets, DeepAR generates probabilistic forecasts with the lowest $\mathrm{nMPI}(\beta)$s. However, for the Solar data set, the $\mathrm{nMPI}(\beta)$s from the QRNN are the smallest. The Bike data set is interesting for the benchmarks since the $\mathrm{nMPI}(\beta)$ with $1-\beta = 98\%$ is the smallest for DeepAR, but the $\mathrm{nMPI}(\beta)$ with $1-\beta = 70$ is the smallest for the QRNN.

Prediction Intervals To evaluate calibration and sharpness simultaneously, we consider the quality of the prediction intervals generated with our approach and the direct probabilistic benchmarks. For this purpose, we report the average MW score across five runs in Table 4.18. We first observe that our approach results in the lowest MW scores for every data set. Furthermore, the MW scores for each point forecaster, when combined with the cINN, are lower than any of the direct benchmarks on all data sets. Regarding the direct probabilistic benchmarks, the best-performing model depends on the data set considered. DeepAR results in the lowest MW scores for the Electricity, Price, and Bike data sets, whilst QRNN results in the lowest MW scores for the Solar data set.

Table 4.18.: Comparison of the average MW score between the probabilistic forecasts from the cINN and the direct probabilistic benchmarks. The average MW score is calculated over five runs on the test data set, and the best values for each data set are highlighted in bold. [3]

Data Set	Electricity	Price	Solar	Bike
XGBoost-cINN	**19.4470**	13.6459	**7.3523**	35.4601
N-HiTS-cINN	25.7098	12.5044	11.1579	27.2643
TFT-cINN	22.1819	**11.1381**	7.9020	**22.0296**
DeepAR	28.2175	17.6831	16.3384	36.7610
QRNN	32.2938	20.7876	15.4228	46.4754
NNQF	29.4266	24.5482	16.7298	47.0819

Qualitative Analysis

As a final comparison to the benchmarks, we qualitatively compare prediction intervals and calibration for the Price data set to gain further insight into the characteristics of probabilistic forecasts generated by our approach and the considered benchmarks. In this analysis, we only consider the Conformal PI from the first group of benchmarks since this method performs overall best compared to the other benchmarks in that group.

We plot the 98%, 70%, and 40% prediction intervals for a single day in the test data set in Figure 4.2. Compared to the Conformal PI, our approach generates probabilistic forecasts with the narrowest prediction intervals regardless of the base point forecaster used. In fact, for probabilistic forecasts generated with the N-HiTS or TFT base point forecaster, our approach using a cINN results in the narrowest prediction intervals overall. Furthermore, whilst the 40% and 70% Conformal PIs are only slightly wider than those generated by the cINN, the 98% prediction intervals are by far the widest of all considered benchmarks. The three direct probabilistic benchmarks generate prediction intervals that are generally wider than those generated by the cINN but narrower than the Conformal PIs.

To further analyse the calibration of our forecasts, we plot the forecast quantile coverage against the theoretical quantile coverage as a calibration plot in Figure 4.3. We observe that, for all base point forecasters, the Conformal PI provides the most calibrated forecasts, with hardly any deviation from the diagonal. However, our approach using a cINN also results in forecasts that only slightly deviate from the diagonal by slightly overestimating the lower quantiles and slightly underestimating the upper quantiles. From the direct probabilistic benchmarks, the NNQF achieves similar results to our approach using a cINN, whilst the results of DeepAR and the QRNN are noticeably worse.

4.5 Discussion

In this section, we first discuss the forecasting performance of our approach and the associated implications before we highlight some of the key insights gained from the evaluation.

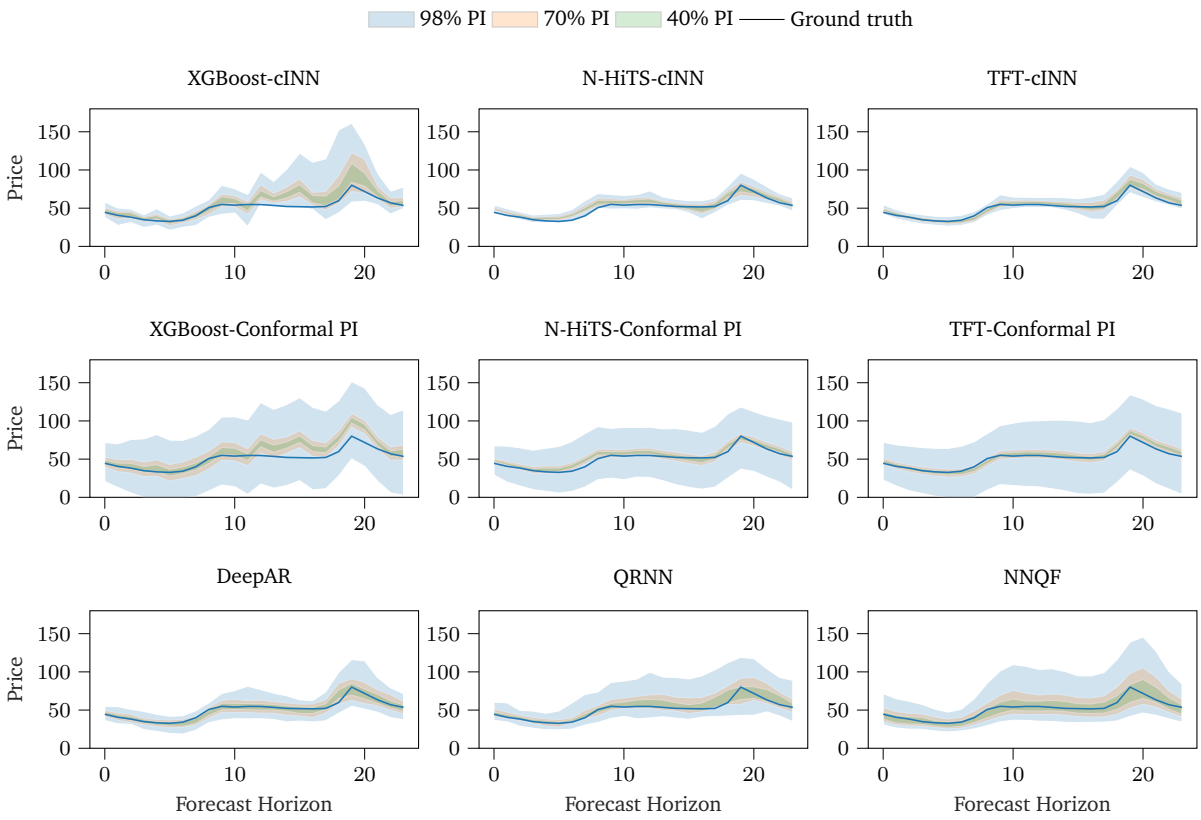


Figure 4.2.: Exemplary 98%, 70%, and 40% prediction intervals on the 11.12.2013 for the Price data set. Probabilistic forecasts are generated by using XGBoost, N-HiTS, and the TFT as base point forecasters and either combining them with our cINN or applying Conformal PI. Further, we compare the three direct probabilistic benchmarks: DeepAR, QRNN, and NNQF. [3]

4.5.1 Forecasting Performance

With regard to forecasting performance, we first discuss the performance of our approach with different point forecasters before comparing our approach to other probabilistic benchmarks. When comparing different point forecasters in our approach, we note that, in most cases, the most accurate point forecasts result in the highest quality probabilistic forecasts when combined with the cINN. This observation is unsurprising since the cINN in our approach includes uncertainty around the initial point forecast and, therefore, the more accurate the point forecast is, the easier it is to effectively include uncertainty.

When comparing our approach to the selected benchmarks, we make several observations. First, our approach almost always outperforms all benchmark models regarding CRPS. In the few occasions where our approach does not result in the lowest CRPS, the difference between the best performing benchmark is small. Second, our approach is not optimally calibrated. Although the forecasts generated with our cINN are well calibrated compared to the direct probabilistic benchmarks, the Empirical PI and Conformal PI achieve lower MAQDs on all data sets. However, it is worth noting that prediction interval-based approaches are specifically designed to achieve certain coverage levels, and further, considering the calibration plots in Figure 4.3 suggests that the difference in calibration may not be as noticeable as the raw MAQD numbers suggest. Third,

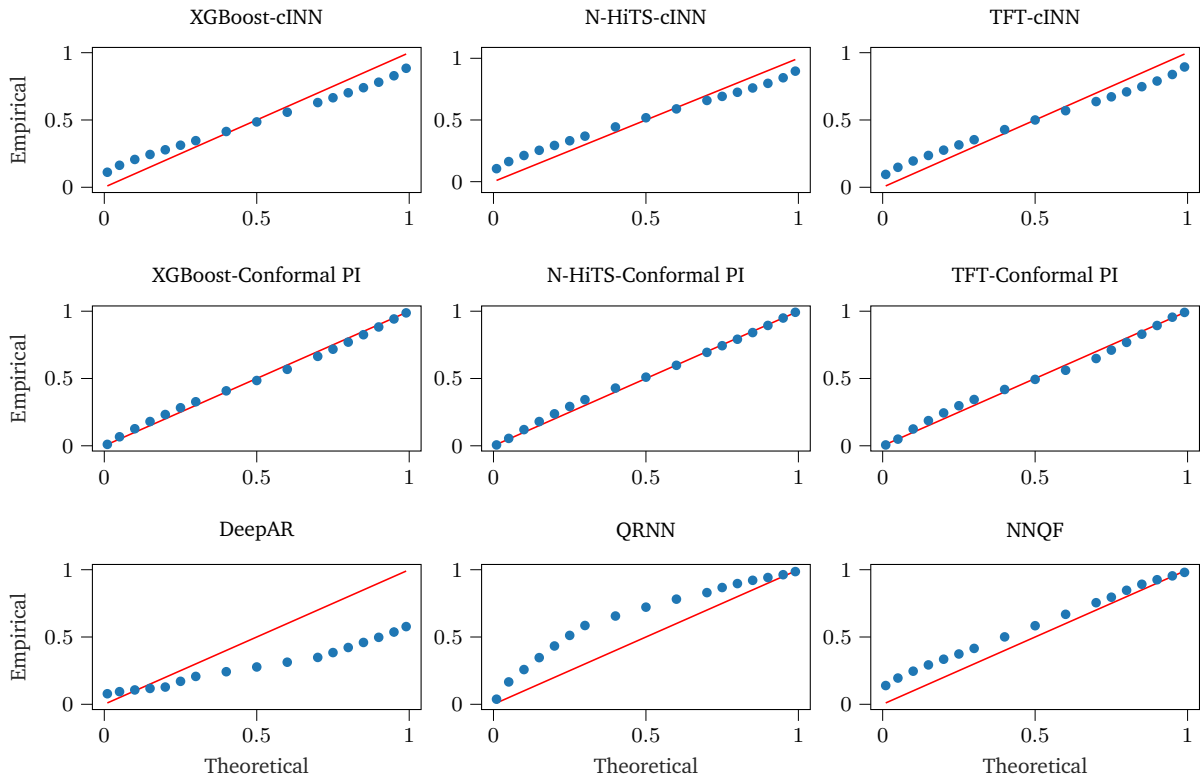


Figure 4.3.: Exemplary calibration plots comparing the theoretical and forecast quantiles on the Price data set, with the red diagonal indicating zero deviation. We compare probabilistic forecasts generated by using XGBoost, N-HiTS, and the TFT as base point forecasters and either combining them with our cINN or applying Conformal PI. Further, we compare the three direct probabilistic benchmarks: DeepAR, QRNN, and NNQF. [3]

our approach consistently generates the sharpest probabilistic forecasts with the lowest $\mathrm{nMPI}(\beta)$. This observation is further highlighted by Figure 4.2 where the forecasts from the N-HiTS and TFT base point forecaster combined with the cINN are far narrower than those of any other benchmarks. Fourth, our approach outperforms all considered benchmarks on all data sets with regards to MW scores. Since Winkler scores value sharp forecasts, this result is not surprising since our approach generates forecasts with narrow prediction intervals. Furthermore, in Figure 4.2 we observe that although the prediction intervals of our approach are narrow, the ground truth is still almost always contained within the interval. In comparison, the other benchmark methods, specifically the prediction interval-based approaches, appear to overestimate the width of the prediction intervals, which adversely affects the Winkler score. Finally, a key takeaway from our evaluation is that both our approach and each of the considered benchmarks have strengths and weaknesses. Whilst our approach results in narrow prediction intervals and low CRPS scores, this comes at the cost of calibration performance. In contrast, the prediction interval-based benchmarks are highly calibrated but generate far wider prediction intervals which results in a worse performance with regard to Winkler scores. Therefore, the best probabilistic forecast may vary, depending on the requirements of the considered situation.

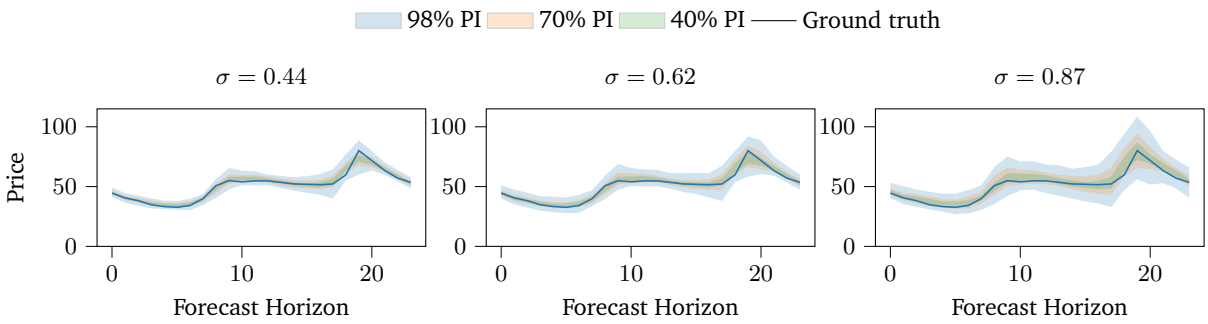


Figure 4.4.: Exemplary prediction intervals using the TFT on the Price data set. We show how the sampling hyperparameter σ affects the resulting probabilistic forecast. The best-performing sampling hyperparameter for the TFT is $\sigma = 0.62$, and the alternate values of σ include varying amounts of uncertainty.

4.5.2 Insights

In addition to the results, there are a few insights regarding the sampling in the latent space and the flexible nature of our approach, which we discuss here.

Sampling in Latent Space Our approach includes uncertainty in point forecasts via latent space distribution sampling. Currently, this sampling is performed by adding normally distributed random noise $\mathbf{r}^i \sim \mathcal{N}(0, \sigma)$ to the point forecast. This approach has several limitations. First, the sampling hyperparameter σ is manually selected to generate optimal forecasts according to CRPS. However, by varying this sampling hyperparameter, it is possible to generate different probabilistic forecasts which follow the same general shape but vary in the amount of uncertainty considered, as illustrated for an exemplary prediction interval in Figure 4.4. Therefore, it may be interesting to investigate methods to automatically select an optimal sampling hyperparameter given the observed data, a selected base forecaster, and a specific evaluation metric. Second, the current approach to optimise σ is rather rudimentary and based on a single evaluation metric. Therefore, it would be interesting to adapt this optimisation, perhaps by adapting concepts from conformal prediction to calculate the nonconformity scores of the samples. Furthermore, optimising the samples based on the resulting quantiles used as an output of the probabilistic forecast might be interesting. With such a strategy, Bonferroni correction [175] could possibly be applied to improve the calibration of our approach.

Flexible Nature In the present chapter, we evaluate the probabilistic forecasting performance of a single selected base point forecaster combined with the cINN. However, our approach is independent of the base point forecast considered, i.e. once the cINN has been trained for a given data set, we can generate probabilistic forecasts from any arbitrary point forecast without retraining. This is advantageous compared to other methods using cINNs or GANs, which require the generative model to be retrained whenever the point forecast is altered. Moreover, such an approach allows us to easily generate an ensemble of probabilistic forecasts based on different

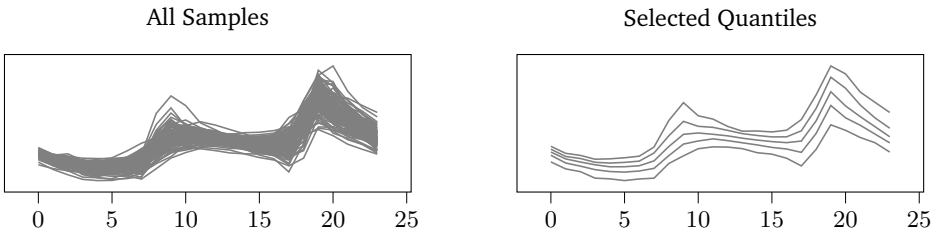


Figure 4.5.: Our approach first generates samples and we calculated the desired quantiles from these samples. Here, we visualise exemplary samples and selected quantiles for a single forecast when using the TFT on the Price data set.

point forecasts. Furthermore, for similar data sets, it may be possible to generate probabilistic forecasts with a generalised cINN that is only trained once on all data sets or a subset thereof.

Another important aspect is that our approach is not limited to prediction intervals or specific quantiles. Whilst our approach currently outputs quantiles based on samples from the latent space, the generative nature of the cINN enables us to generate an arbitrary number of samples and either use these directly to form an ensemble forecast or to output an empirical forecast distribution. The internal workings of our approach are illustrated in an exemplary manner in Figure 4.5. The raw samples are contained in $\hat{\mathscr{Y}}^{\sigma}_{t+H}$ and represent the uncertainty, and from $\hat{\mathscr{Y}}^{\sigma}_{t+H}$ we can easily extract an arbitrary number of samples, or even approximate the distribution. This is advantageous compared to other probabilistic forecast methods that are, by nature, limited to generating prediction intervals.

4.6 Conclusion

In the present chapter, we introduce an approach to generate probabilistic forecasts from arbitrary point forecasts by using a Conditional Invertible Neural Network (cINN) to learn the underlying distribution of the time series data. Our approach maps the underlying distribution of the data to a known and tractable distribution before combining the uncertainty from this known and tractable distribution with an arbitrary point forecast to generate probabilistic forecasts. Importantly, the cINN is independent of the considered point forecast and must not be retrained when the point forecast is altered.

We evaluate our approach by combining multiple point forecasts with a cINN and comparing the resulting probabilistic forecasts with six probabilistic benchmarks on four data sets. We show that our approach generally outperforms all benchmarks regarding Continuous Ranked Probability Score (CRPS) and Winkler scores. Further, our approach generates probabilistic forecasts with the narrowest prediction intervals whilst maintaining reasonable performance in calibration.

Our approach offers a solution to generate flexible probabilistic forecasts based on arbitrary point forecasts. In future work, this flexibility should be further investigated by developing a more advanced strategy for selecting the sampling hyperparameter to improve the calibration

of our probabilistic forecasts. Furthermore, automating the selection of this sampling hyperparameter and considering how different metrics for optimising this parameter affect the resulting forecasts should be investigated. Finally, it may be interesting to explore the performance of our approach using a generalised cINN to generate probabilistic forecasts on multiple data sets without retraining.

5 Customising the Properties of Probabilistic Forecasts

The content of this chapter is based on:

K. Phipps *et al.*, "Loss-customised probabilistic energy time series forecasts using automated hyperparameter optimisation", in *Proceedings of the Fourteenth ACM International Conference on Future Energy Systems*, ACM, 2023, pp. 271–286. DOI: 10.1145/3575813.3595204.

In the previous chapter, we introduced an approach to generating probabilistic forecasts from arbitrary point forecasts. However, one of the key observations was that neither our method nor the considered benchmarks were best for all evaluation metrics considered. For example, whilst our method resulted in probabilistic forecasts with the narrowest prediction intervals and best Mean Winkler (MW) scores, it performed worse in terms of calibration, i.e. Mean Absolute Quantile Deviation (MAQD). Furthermore, probabilistic forecasts are usually generated for use in a certain application, and these applications often require specific probabilistic forecast properties [176]. For example, whilst a stochastic optimisation problem generally requires an accurate representation of the uncertainty in all future scenarios, a robust optimisation problem may only be interested in extreme events occurring with low probability. As a result, stochastic optimisation requires probabilistic forecasts that are sharp enough to convey information whilst being reasonably calibrated, while robust optimisation requires probabilistic forecasts that weigh calibration higher.

Unfortunately, existing probabilistic forecasting methods generate probabilistic forecasts whose properties cannot be easily customised [170], [176]–[179]. If the properties of this probabilistic forecast do not fit the requirements of the downstream application, for example, the forecast is too sharp, then an alternative forecast method must be selected, which requires potentially computationally expensive retraining. However, one of the main advantages of our probabilistic forecasting approach introduced in Chapter 4 is that the sampling hyperparameter influences the characteristics of the resulting probabilistic forecast and, additionally, can be selected to minimise an arbitrary probabilistic loss metric.

Therefore, in the present chapter, we extend the approach from Chapter 4 by using automated Hyperparameter Optimisation (HPO) to select the sampling hyperparameter σ based on custom probabilistic loss metrics. By considering custom probabilistic loss metrics that focus on different characteristics, this automated HPO generates loss-customised probabilistic forecasts with different properties. Since the HPO to determine an optimal $\sigma^\star$ only depends on the previously generated latent space representation of the forecast, we can apply custom loss metrics to

optimise the hyperparameter in an automated manner and generate different, loss-customised probabilistic forecasts without retraining the used base point forecaster or the cINN.

In this chapter, we present the automated HPO extension that enables us to generate loss-customised probabilistic forecasts without computationally expensive retraining and empirically evaluate this approach by creating loss-customised probabilistic forecasts based on six different loss metrics for four real-world data sets. The rest of the present chapter is structured as follows. In Section 5.1, we present work related to our approach and highlight the identified research gap and our specific contribution. We then present the extension to our approach introduced in Chapter 4, which generates loss-customised probabilistic forecasts in Section 5.2, before introducing the experimental setting used for the evaluation in Section 5.3. We report the results of our evaluation in Section 5.4 and discuss these results in Section 5.5. Finally, we conclude and propose future research directions in Section 5.6.

5.1 Related Work

Since the extension introduced in this chapter generates loss-customised probabilistic forecasts using automated HPO, we consider related work in the field of customised probabilistic forecasts and automated probabilistic forecasts. Finally, we highlight the identified research gap and our specific contribution.

Customised Probabilistic Forecasts Whilst extensive research exists on probabilistic forecasts for time series, for example, load [180]–[182], electricity price [183], [184], wind power generation [185], [186], solar power generation [187], [188], and mobility [189], these existing methods do not provide forecasts with customised properties. Specifically, each method results in a probabilistic forecast whose properties cannot be altered by the probabilistic forecasting method once training is complete. On a more general level, even modern probabilistic forecasting methods aiming to be flexible by applying non-parametric regression splines [179] or temporal convolution neural networks [178] still result in probabilistic forecasts whose properties cannot be easily customised. Although it may be possible to alter such forecasts with post-processing techniques, such techniques usually rely on estimating a distribution for the target time series and applying statistical methods to approximate this distribution [109], [190] or applying non-parametric deep learning methods [73], [191].

Automated Probabilistic Forecasts Although extensive research exists regarding automated point forecasts, almost no work focuses on automated probabilistic forecasts [9], [192]. In [193], an end-to-end learning approach enables coherent probabilistic hierarchical time series forecasts. However, this approach is not fully automated, only focuses on hierarchical time series forecasting, and is not designed for individual time series with periodicities and seasonality. Furthermore, a recent software package from Shchur *et al.* [194], provides a framework for

automating the hyperparameter optimisation of probabilistic forecasting models. Whilst this framework is promising, it focuses on the hyperparameters of existing models and has not yet been extensively evaluated for specific use cases. With regards to point forecasts, Zhao *et al.* [195] use automation to generate the most suitable point ensemble prediction model for solar power forecasting, whilst Meisenbacher *et al.* [196] use automation to optimise pre-trained point models and their contribution to the ensemble. Similarly, in [197], automation is applied in the design process of the point solar irradiance forecast to optimise the neural network architecture.

Research Gap & Specific Contribution Based on the reviewed literature, we identify two clear research gaps. First, to the best of our knowledge, no method exists that generates probabilistic forecasts whose properties can be customised without retraining. More specifically, no method can generate multiple, different, probabilistic forecasts without retraining the underlying forecast model. Second, to the best of our knowledge, almost no work exists that combines automation with probabilistic time series forecasts. Only Shchur *et al.* [194] present a framework to enable this combination, but their framework has not yet been extensively applied or evaluated. Therefore, the specific contribution of our approach is a method that is capable of generating probabilistic forecasts whose properties can be customised without retraining the underlying forecast model. Furthermore, since this customisation is achieved through an automated HPO, a further contribution is the combination of automation with probabilistic time series forecasts and the evaluation of this combination on multiple real-world data sets.

5.2 Loss-Customised Probabilistic Forecasts

To generate loss-customised probabilistic forecasts, we combine our approach from Chapter 4 [3] with automated HPO. This combination is shown in Figure 5.1, with the forecasting approach illustrated by the diagram and the automated HPO highlighted in blue. This section explains how our approach from Chapter 4 is combined with automated HPO to generate loss-customised forecasts. We first recap the setting from the previous chapter before detailing how the sampling hyperparameter influences the properties of the probabilistic forecast and describing how we include automated HPO based on custom loss metrics to optimise this parameter.

Recap: Generating Probabilistic Forecasts As a reminder, our approach from Chapter 4 generates probabilistic forecasts by analysing the neighbourhood of the representation $\hat{\mathbf{z}}_{t+H}$ of a point forecast in the Gaussian distributed latent space. Thereby, our forecasting approach explores the uncertainty in the neighbourhood of the forecast with

$$\tilde{\mathbf{z}}^i_{t+H} = \hat{\mathbf{z}}_{t+H} + \mathbf{r}^i, \quad i = 1, \dots, I, \ \mathbf{r}^i \sim \mathcal{N}(0, \sigma), \tag{5.1}$$

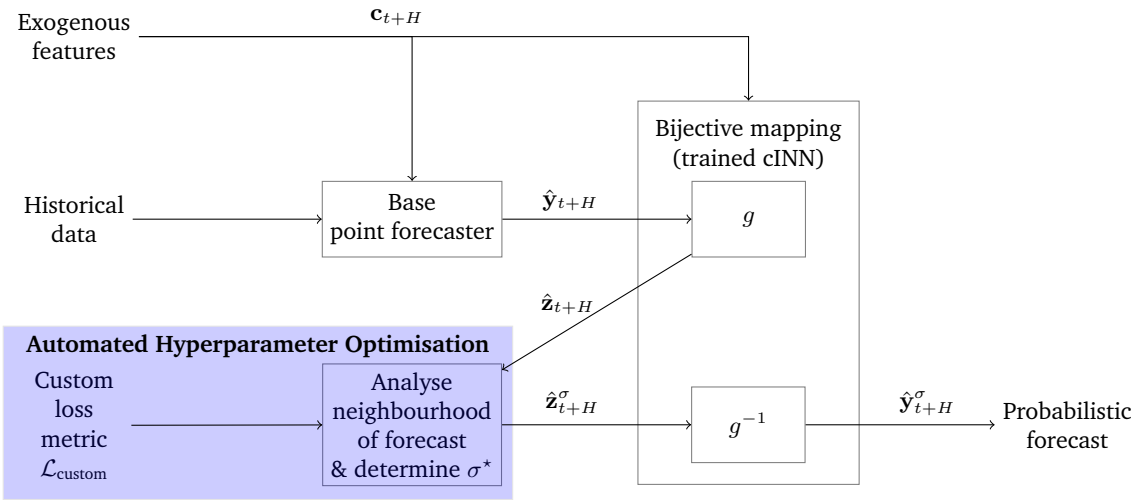


Figure 5.1.: Overview of the approach to generate loss-customised probabilistic forecasts. As in Chapter 4, exogenous features and historical data are used as inputs by a base point forecaster to generate a point forecast in an unknown distribution. This point forecast and the exogenous features are combined in a Conditional Invertible Neural Network (cINN), which generates a representation of the forecast in a known and tractable distribution. The neighbourhood of this representation is analysed to determine how to include uncertainty information using an automated HPO so that it optimises a custom loss metric. Finally, with a backwards pass through the cINN, the uncertainty is mapped back to the unknown distribution to generate the customised probabilistic forecast. [4]

where $\mathbf{r}^i$ is a random noise taken from a standard normal distribution with mean 0 and variance σ, and $i = 1, \ldots, I$ is the sample number.[1] By sampling I different random noises and adding these to $\hat{\mathbf{z}}_{t+H}$, a tuple of realisations

$$\hat{\mathbf{z}}^{\sigma}_{t+H} = \left(\tilde{\mathbf{z}}^{i}_{t+H}\right)_{i=1,\cdots,I} \tag{5.2}$$

is generated, which is dependent on the sampling variance σ and similar but not identical to the original forecast, thus representing the uncertainty in the neighbourhood. To generate a probabilistic forecast, we then use the inverse mapping of the cINN to map the uncertainty in the Gaussian distributed space back to the original space with an unknown distribution. Finally, in the present chapter, we calculate the quantiles of the resulting samples to generate a probabilistic forecast $\hat{\mathbf{y}}^{\sigma}_{t+H}$, although it is possible to use the resulting samples directly as an ensemble probabilistic forecast.

Effect of the Sampling Hyperparameter The sampling parameter σ in our forecasting approach is responsible for introducing uncertainty information into the forecast. Depending on how σ is selected, the properties of the resulting probabilistic forecast change. For example, a small σ only allows for a small amount of uncertainty to be included and will result in sharp probabilistic forecasts. In contrast, a large σ includes more uncertainty, which could result in better calibration. Therefore, this single sampling hyperparameter drastically influences the

[1] In this case, $\mathcal{N}(0, \sigma)$ is a H-Dimensional normal distribution, where σ is the same for each dimension.

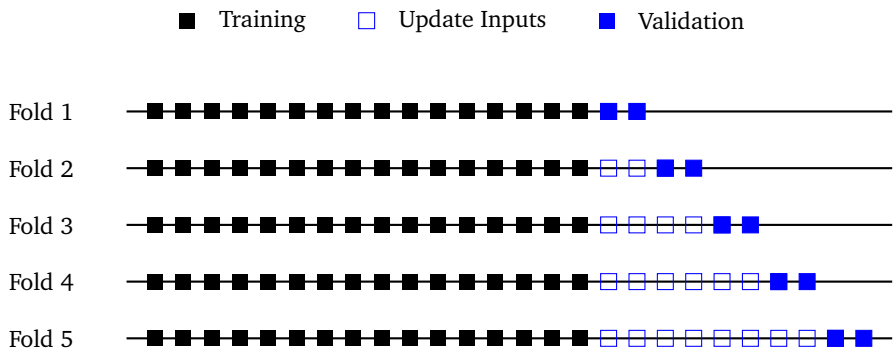


Figure 5.2.: Visualisation of a five-fold rolling origin update time series CV based on [198]. Each block indicates a multi-horizon forecast $\hat{\mathbf{y}}_{t+H}$ with forecast horizon H. The k folds in the validation set are sequential to preserve the temporal dependencies of the time series. The rolling origin update CV allows us to determine $\sigma^\star$ independently from the training of the base point forecaster and cINN whilst including up-to-date inputs, avoiding retraining. [4]

properties of the resulting probabilistic forecast, and, dependent on the desired properties of the forecast, a different σ will be optimal.

Automated Optimisation of the Sampling Hyperparameter Although the sampling hyperparameter influences the properties of the resulting probabilistic forecast, determining which σ is optimal is not trivial. Instead of manually searching for a σ that results in probabilistic forecasts with the desired characteristics, we propose optimising σ based on a selected custom loss metric $\mathcal{L}_{\text{custom}}$ in an automated manner. Thereby, we apply an automated HPO to search for

$$\sigma^\star = \inf \arg\min_{\sigma \in \Sigma} \mathcal{L}_{\text{custom}} \left(\hat{\mathbf{y}}^{\sigma}_{t+H}, \mathbf{y}_{t+H} \right), \tag{5.3}$$

where Σ is the set of considered sampling hyperparameters. Importantly, this automated HPO can be easily applied with any defined custom loss metric $\mathcal{L}_{\text{custom}}$. Therefore, as long as a loss metric that supports the desired properties of the resulting probabilistic forecast is selected, the selected $\sigma^\star$ generates a loss-customised probabilistic forecast with exactly these desired properties.

When applying our automated HPO, we select $\sigma^\star$ by calculating $\mathcal{L}_{\text{custom}}$ on a validation data set to prevent overfitting. Furthermore, to ensure a stable automated HPO, we determine $\sigma^\star$ based on the mean $\mathcal{L}_{\text{custom}}$ during a k-fold Cross Validation (CV). As the CV method, we apply the rolling origin update time series CV, visualised in Figure 5.2. The idea behind the rolling origin update CV is that the k-folds of the CV are sequential. Specifically, we divide the validation data set into k sequential folds and perform the CV by moving forward one fold for each CV iteration. Furthermore, when we move forward a fold, we update the inputs to include information from the most recent fold. This update involves simply considering the measured time series inputs from the most recent fold as inputs for the next fold. As a result, the rolling origin update CV ensures the temporal dependencies of the time series are considered, and the forecast horizon always remains the same [198]. Furthermore, a rolling origin update approach enables us to apply our automated HPO without retraining the base point forecaster or cINN whilst ensuring up-to-date inputs are available.

Integration Into the Forecasting Approach To generate these loss-customised probabilistic forecasts, we apply the described automated HPO in the third step of the applied forecasting approach, i.e. analysing the neighbourhood (see Figure 5.1). Since finding $\sigma^\star$ is independent of the first and second steps, computationally expensive retraining of the base point forecaster and the cINN is not required. As a result, combining the applied forecasting approach with automated HPO allows us to customise the properties of the resulting probabilistic forecast in an automated manner simply by varying the selected loss metric $\mathcal{L}_{\text{custom}}$ and restarting the automated HPO to find the corresponding $\sigma^\star$.

5.3 Experimental Setting

To evaluate the loss-customised forecasts generated by our approach, we consider an identical experimental setting to the previous chapter, described fully in Section 4.3. We use the same four data sets: Electricity, Price, Solar, and Bike, consider the same six base point forecasters, and use the same architecture for the cINN. Furthermore, we consider the same evaluation metrics as the previous chapter, i.e. the CRPS (see Equation (2.12)), the MAQD (see Equation (2.20)), the normalised Mean β-PI Width ($\text{nMPI}(\beta)$) (see Equation (2.18)), and the MW (see Equation (2.17)). Therefore, in this section, we present the additions to the experimental setting, i.e. the applied loss metrics for customising the probabilistic forecasts and the configuration of the HPO.

5.3.1 Loss Metrics for the Automated Hyperparameter Optimisation

We apply six loss metrics for the automated HPO. The first metric is the mean Pinball Loss (PL) introduced in Equation (2.15).[2] The mean PL considers each quantile individually and, similar to the CRPS accounts for both sharpness and calibration. The second loss metric is the MW, which we described in Equation (2.17). The MW focuses on the quality of prediction intervals and rewards sharp forecasts. The third loss metric is the MAQD introduced in Equation (2.20), which focuses on rewarding a well-calibrated forecast. In addition to these three loss metrics, we introduce three further custom metrics in the following. The fourth loss metric is the Coverage Rate Error (CRE), described in Equation (2.22), which is designed to optimise the Coverage Rate (CR) of the customised forecast. In the present chapter, we perform loss-customisation with the CRE by considering $\mathcal{B} = \{(0.01, 0.99), (0.05, 0.95), (0.15, 0.85), (0.25, 0.75), (0.3, 0.7), (0.4, 0.6)\}$. The fifth loss metric is the Extreme Quantile Deviation (EQD) introduced in Equation (2.23), which only considers extreme quantiles and should lead to probabilistic forecasts with wide prediction intervals accounting for extreme regions of uncertainty. The last loss metric considered for the customisation is the Upper Quantile Deviation (UQD), described in Equation (2.24). We apply to UQD to investigate whether the assumption that the quantiles are symmetrical around the median holds and whether we can obtain probabilistic forecasts with similar properties to

[2] Note that in the previous chapter, we optimised the forecasts with regards to CRPS. Therefore, we do not consider this loss metric again in the present chapter.

those optimised with the MAQD by only considering half of the quantiles. Furthermore, only considering the upper quantiles should ensure that the uncertainty around peaks in the time series forecast is well approximated.

5.3.2 Configuration of the Automated Hyperparameter Optimisation

The configuration of the applied automated HPO is identical for all considered data sets. To realise the automated HPO, we use the Ray Tune [199] library, which implements the Tree of Parzen Estimators [200] search algorithm. For each data set, we perform the automated HPO for each of the five considered loss metrics. For each of these loss metrics, we consider the search space of $\sigma \in \mathcal{U}(0.1, 2)$, stop the automated HPO with a tuner timeout after $120\,\mathrm{s}$, allow a maximum of eight concurrent trials, and select $k = 5$ as the number of folds for the CV.

5.4 Evaluation

To evaluate the effect of our approach to customise probabilistic forecasts with automated HPO and different probabilistic loss metrics, we conduct a three-step evaluation. First, we compare the customised probabilistic forecasts by analysing quality, calibration, sharpness, and prediction intervals. Second, we qualitatively compare the probabilistic forecasts by plotting prediction intervals and calibration plots. Finally, we analyse the effects of the hyperparameter optimisation.

5.4.1 Quantitative Comparison of Customised Probabilistic Forecasts

In this section, we compare the customised probabilistic forecasts by analysing their quality, calibration, sharpness and prediction intervals.

Quality To compare the quality of the loss-customised probabilistic forecasts generated by using different loss metrics, we report the average CRPS across three evaluation runs in Table 5.1. We observe for all data sets and all base point forecasters, using the mean PL as the custom loss metric in the HPO generates probabilistic forecasts with the lowest CRPS. Furthermore, no other custom loss metric clearly results in the second lowest CRPS when applied in the HPO. The second lowest CRPS can be achieved with either the MW, MAQD, CRE or UQD as the custom loss metric in the HPO, depending on the data set and base point forecaster considered. Only the EQD as the custom loss metric applied in the HPO never results in the second lowest CRPS.

Table 5.1.: The average CRPS across three evaluation runs for all point forecasters combined with the cINN and customised with different loss metrics. The CRPS is calculated on the test set, and the best value for each point forecaster on each data set is marked bold.

Data	Point Forecaster	Custom Loss Metrics					
		PL	MW	MAQD	CRE	EQD	UQD
Electricity	LR	**0.3288**	0.3697	0.5802	0.4876	0.5738	0.3585
	RF	**0.2367**	0.2722	0.2926	0.2574	0.2970	0.2375
	NN	**0.2575**	0.2966	0.3410	0.2848	0.3492	0.2582
	XGBoost	**0.2377**	0.2611	0.2941	0.2631	0.3159	0.2383
	N-HiTS	**0.2937**	0.3386	0.3320	0.3345	0.3943	0.3175
	TFT	**0.2786**	0.3229	0.3061	0.3031	0.3364	0.2837
Price	LR	**0.1677**	0.2035	0.1745	0.1748	0.1829	0.1735
	RF	**0.1721**	0.2095	0.1820	0.1892	0.1980	0.1763
	NN	**0.1734**	0.2096	0.1893	0.1900	0.2139	0.1884
	XGBoost	**0.1589**	0.1865	0.1677	0.1686	0.1873	0.1679
	N-HiTS	**0.1544**	0.1869	0.1641	0.1688	0.1839	0.1614
	TFT	**0.1457**	0.1776	0.1565	0.1560	0.1659	0.1531
Solar	LR	**0.1662**	0.1719	0.4516	0.4651	0.5413	0.3959
	RF	**0.1054**	0.1236	0.1065	0.1054	0.1115	0.1067
	NN	**0.1381**	0.1488	0.1961	0.2183	0.2520	0.1850
	XGBoost	**0.1069**	0.1229	0.1071	0.1070	0.1117	0.1071
	N-HiTS	**0.1603**	0.1684	0.2697	0.2577	0.2590	0.2810
	TFT	**0.1202**	0.1385	0.1393	0.1307	0.1377	0.1387
Bike	LR	**0.4520**	0.5344	0.5410	0.5243	0.5334	0.5482
	RF	**0.4961**	0.6393	0.5371	0.5209	0.5377	0.5370
	NN	**0.4617**	0.5573	0.5727	0.5008	0.5219	0.5735
	XGBoost	**0.4604**	0.5427	0.5473	0.5041	0.5258	0.5552
	N-HiTS	**0.3555**	0.3791	0.4566	0.4531	0.5282	0.4492
	TFT	**0.2714**	0.2865	0.3160	0.3139	0.3722	0.3182

Calibration To compare the calibration of the loss-customised forecasts generated by applying different loss metrics to the automated HPO, we report the average MAQD across three runs in Table 5.2. The first observation is that using the MW as the loss metric to generate loss-customised probabilistic forecasts always results in the highest MAQD. Noticeably, the MAQD from probabilistic forecasts generated based on the MW is often orders of magnitude larger than the MAQD resulting from the other loss metrics. Second, the PL also never achieves the lowest MAQD, however these MAQDs results are of a similar order in magnitude to those from the calibration-based loss metrics. Specifically, unlike the MW customised forecasts, the forecasts customised using the PL are not noticeably worse. Third, using the MAQD to customise the probabilistic forecasts does not always result in probabilistic forecasts with the lowest MAQD. Although using the MAQD as a custom loss metric results in the best calibration 13 times, using the CRE, EQD, and UQD results in the best calibration three or four times for each loss metric. Therefore, the choice of loss metric to achieve the best calibration, i.e. the lowest MAQD, is dependent on the base point forecaster and data considered.

Sharpness To compare the sharpness of our loss-customised forecasts generated with different loss metrics, we report the average $\mathrm{nMPI}(\beta)$ across three runs for $1-\beta = 98\%$ and $1-\beta = 70\%$ in Table 5.3. The first observation is that using the MW as a loss metric to customise the

Table 5.2.: The average MAQD across three evaluation runs for all point forecasters combined with the cINN and customised with different loss metrics. The MAQD is calculated on the test set and the best value for each point forecaster on each data set is marked bold.

Data	Point Forecaster	Custom Loss Metrics					
		PL	MW	MAQD	CRE	EQD	UQD
Electricity	LR	0.1171	0.2727	0.0533	0.0604	**0.0531**	0.0879
	RF	0.0695	0.2592	0.0452	0.0504	**0.0447**	0.0855
	NN	0.0716	0.2469	**0.0389**	0.0459	0.0426	0.0793
	XGBoost	0.0447	0.2200	**0.0347**	0.0229	0.0444	0.0499
	N-HiTS	0.0516	0.2580	**0.0176**	0.0182	0.0341	0.0189
	TFT	0.0386	0.2596	**0.0142**	0.0131	0.0296	0.0254
Price	LR	0.0371	0.2614	0.0311	0.0309	0.0360	**0.0303**
	RF	0.1009	0.2803	0.0915	**0.0908**	0.0913	0.0938
	NN	0.0988	0.2754	**0.0960**	0.0962	0.0971	0.0961
	XGBoost	0.0767	0.2449	0.0732	0.0730	0.0743	**0.0729**
	N-HiTS	0.0772	0.2641	**0.0494**	0.0508	0.0576	0.0503
	TFT	0.0570	0.2694	0.0171	**0.0160**	0.0228	0.0219
Solar	LR	0.1688	0.2328	0.0221	0.0223	**0.0210**	0.0292
	RF	0.0733	0.1164	**0.0670**	0.0679	0.0914	0.0672
	NN	0.1394	0.2322	**0.0341**	0.0345	0.0342	0.0375
	XGBoost	0.0227	0.2136	**0.0215**	0.0219	0.0464	0.0222
	N-HiTS	0.1667	0.2479	**0.0426**	0.0450	0.0450	0.0427
	TFT	0.0634	0.2486	0.0305	**0.0157**	0.0319	0.0247
Bike	LR	0.1135	0.2703	**0.0466**	0.0492	0.0479	0.0467
	RF	0.1735	0.3358	**0.1132**	0.1259	0.1133	0.1134
	NN	0.1386	0.2963	**0.0784**	0.0970	0.0868	0.0785
	XGBoost	0.1114	0.2686	0.0572	0.0696	0.0615	**0.0571**
	N-HiTS	0.0839	0.2303	0.0279	0.0266	0.0455	**0.0259**
	TFT	0.0515	0.2193	0.0265	**0.0249**	0.0494	0.0277

probabilistic forecast always results in the sharpest probabilistic forecasts. Second, we observe that the probabilistic forecasts customised with the PL generally result in probabilistic forecasts with the second smallest $\mathrm{nMPI}(\beta)$. Third, all other loss metrics result in probabilistic forecasts that generally have a larger $\mathrm{nMPI}(\beta)$. The magnitude of this difference depends on the selected data set and the base point forecaster. For example, on the Price data set, the UQD as a custom loss metric results in a lower $\mathrm{nMPI}(\beta)$ for the random forest base point forecaster than the PL. However, on the Bike data set, the MAQD, CRE, and EQD as custom loss metric results in probabilistic forecasts with noticeably larger $\mathrm{nMPI}(\beta)$s when compared to the PL.

Prediction Intervals To compare both calibration and sharpness and evaluate the quality of the prediction intervals, we report the average MW across three evaluation runs for all loss metrics used to customise the probabilistic forecasts in Table 5.4. We observe that using the MW as a loss metric to customise the probabilistic forecasts always generates loss-customised probabilistic forecasts with the lowest MW value. Second, we observe that the second lowest MWs values are generally achieved by using the PL as a loss metric to customise the probabilistic forecasts, although sometimes loss-customisation with the UQD performs better. Third, the EQD almost always results in probabilistic forecasts with the highest MW when it is used as a loss metric to customise the probabilistic forecasts. Finally, we observe that the calibration-based loss metrics,

Table 5.3.: The average $\mathrm{nMPI}(\beta)$ for $1-\beta = 98\%$ and $1-\beta = 70\%$ across three evaluation runs for all point forecasters combined with the cINN and customised with different loss metrics. The $\mathrm{nMPI}(\beta)$ is calculated on the test set, and the best value for each point forecaster and each β on each data set is marked bold.

Data	Point Forecaster	PI	Custom Loss Metrics					
			PL	MW	MAQD	CRE	EQD	UQD
Electricity	LR	98%	1.6596	**0.2190**	9.0970	6.3953	8.9233	2.9807
		70%	0.6651	**0.1025**	2.2763	1.7926	2.2423	1.0517
	RF	98%	1.0707	**0.1350**	2.3841	1.8020	2.7019	1.0073
		70%	0.4907	**0.0638**	0.9563	0.7709	1.0382	0.4626
	NN	98%	1.2668	**0.2010**	3.0479	2.0384	3.3326	1.2650
		70%	0.5577	**0.0951**	1.0680	0.8130	1.1291	0.5580
	XGBoost	98%	1.2387	**0.2473**	2.8556	2.0751	3.4465	1.2968
		70%	0.5245	**0.1166**	0.9717	0.7797	1.0988	0.5455
	N-HiTS	98%	1.7322	**0.1914**	3.7476	3.3837	5.2366	2.9702
		70%	0.6964	**0.0903**	1.2177	1.1357	1.5276	1.0385
	TFT	98%	1.7954	**0.1616**	2.8338	2.6760	3.4794	1.8809
		70%	0.7381	**0.0764**	1.0458	0.9988	1.2020	0.7657
Price	LR	98%	0.5951	**0.0628**	0.7785	0.8198	0.9850	0.7381
		70%	0.2533	**0.0296**	0.3154	0.3288	0.3795	0.3022
	RF	98%	0.5838	**0.0435**	0.7617	0.9583	1.0878	0.6668
		70%	0.2489	**0.0205**	0.3091	0.3695	0.4088	0.2777
	NN	98%	0.6152	**0.0543**	0.8412	0.9559	1.4516	0.8173
		70%	0.2638	**0.0256**	0.3413	0.3780	0.5175	0.3331
	XGBoost	98%	0.5946	**0.0881**	0.7953	0.8165	1.0656	0.7691
		70%	0.2529	**0.0416**	0.3217	0.3285	0.4050	0.3137
	N-HiTS	98%	0.4719	**0.0539**	0.7096	0.8527	0.9608	0.6466
		70%	0.2063	**0.0255**	0.2916	0.3366	0.3704	0.2705
	TFT	98%	0.5579	**0.0431**	0.8459	0.7080	0.9501	0.8467
		70%	0.2357	**0.0203**	0.3311	0.2869	0.3612	0.3303
Solar	LR	98%	1.4497	**0.5081**	12.3309	11.7126	10.7218	8.6557
		70%	0.6749	**0.2412**	2.3293	2.2713	2.1978	2.0477
	RF	98%	0.9296	**0.0825**	0.5806	0.7508	1.3528	0.5453
		70%	0.4379	**0.0391**	0.2753	0.3556	0.6295	0.2584
	NN	98%	1.2308	**0.3775**	3.8415	4.8428	6.0429	3.1930
		70%	0.5713	**0.1788**	1.3794	1.5240	1.6820	1.2498
	XGBoost	98%	1.0006	**0.1473**	0.8606	0.8582	1.2420	0.8177
		70%	0.4613	**0.0696**	0.4023	0.4020	0.5616	0.3829
	N-HiTS	98%	1.3476	**0.3453**	8.1097	7.4918	7.9578	9.1927
		70%	0.6082	**0.1639**	1.6926	1.6313	1.6606	1.7678
	TFT	98%	0.8745	**0.1137**	1.5476	1.3012	1.4972	1.6674
		70%	0.4142	**0.0539**	0.7112	0.6078	0.6880	0.7565
Bike	LR	98%	1.4189	**0.2390**	3.0377	2.8143	2.9533	3.3306
		70%	0.6153	**0.1127**	1.1304	1.0677	1.1055	1.2076
	RF	98%	1.4034	**0.1275**	2.5732	2.3183	2.5848	2.5704
		70%	0.5982	**0.0602**	0.9870	0.9079	0.9894	0.9848
	NN	98%	1.4554	**0.2138**	3.1160	2.2604	2.5221	3.3443
		70%	0.6244	**0.1008**	1.1393	0.8977	0.9749	1.2011
	XGBoost	98%	1.4404	**0.2641**	3.0000	2.3328	2.5637	3.1972
		70%	0.6186	**0.1244**	1.1200	0.9196	0.9909	1.1705
	N-HiTS	98%	1.0550	**0.2658**	2.5302	2.7089	4.2693	1.8955
		70%	0.4799	**0.1255**	1.0321	1.0885	1.5356	0.8125
	TFT	98%	0.9143	**0.2337**	1.6019	1.8522	2.6506	1.6007
		70%	0.4201	**0.1103**	0.7042	0.8008	1.0818	0.7035

Table 5.4.: The average MW across three evaluation runs for all point forecasters combined with the cINN and customised with different loss metrics. The MW is calculated on the test set and the best value for each point forecaster on each data set is marked bold.

Data	Point Forecaster	Custom Loss Metrics					
		PL	MW	MAQD	CRE	EQD	UQD
Electricity	LR	37.6814	**9.9407**	182.8969	128.5999	179.2704	56.9971
	RF	22.1695	**6.8959**	51.5464	34.6357	53.7453	19.1363
	NN	25.9474	**8.3779**	76.0516	44.6006	80.8257	24.5457
	XGBoost	25.7299	**8.3805**	58.9752	42.1482	70.5506	24.5363
	N-HiTS	34.8605	**8.7982**	59.4753	60.9907	95.1997	50.9430
	TFT	33.3594	**8.2147**	49.8871	48.3439	66.6108	37.1876
Price	LR	19.2837	**5.1559**	24.0408	24.1874	28.3107	23.3724
	RF	18.2763	**4.8770**	24.5919	28.1990	32.0639	21.4930
	NN	19.6860	**5.1079**	28.1561	28.6118	38.8646	27.7230
	XGBoost	17.4747	**5.3755**	23.1620	23.6462	32.1271	23.2828
	N-HiTS	15.1196	**4.7243**	22.2309	24.5669	31.2884	20.5606
	TFT	15.2615	**4.3300**	22.3699	21.9759	26.7075	20.4656
Solar	LR	15.3591	**7.7364**	239.0324	248.1242	317.3128	187.9659
	RF	9.1180	**2.8953**	6.3459	7.3841	14.0195	6.1668
	NN	12.6392	**5.6797**	56.7435	75.6261	106.2591	48.5392
	XGBoost	9.4543	**3.3421**	9.2688	9.5123	14.6080	8.9439
	N-HiTS	13.9305	**6.1524**	104.8548	92.8597	93.7261	116.0854
	TFT	10.0264	**3.6674**	27.0311	19.3953	24.9879	26.6541
Bike	LR	43.6027	**14.8912**	93.9864	86.1645	90.1602	97.3948
	RF	44.9789	**14.5801**	78.9297	69.0226	79.0359	78.9344
	NN	45.3917	**14.8286**	107.0277	71.2937	82.0552	107.2332
	XGBoost	44.8392	**15.5589**	93.7391	72.9482	83.7189	97.5757
	N-HiTS	36.9596	**13.0882**	81.3684	79.8015	108.8494	79.3144
	TFT	28.8590	**10.0126**	47.6897	46.8275	66.7701	48.5484

i.e. MAQD, CRE, EQD, and UQD perform very inconsistently across all data sets and for all base point forecasters when used for loss-customisation, whilst the performance of the PL and MW is far more consistent.

5.4.2 Qualitative Comparison of Customised Probabilistic Forecasts

To gain further insights into the different probabilistic forecasts generated when customised with various loss metrics, we consider exemplary prediction intervals and calibration plots. We generate these plots for two different point forecasters: the Linear Regression (LR) as the simplest base point forecaster and the Temporal Fusion Transformer (TFT) as the most complex.

Prediction Intervals We plot exemplary 98%, 70%, and 40% prediction intervals for a day on the Price data set for the LR in Figure 5.3, and for the TFT in Figure 5.4. Thereby, we compare the prediction intervals generated when using different loss metrics in the automated HPO to generate loss-customised probabilistic forecasts. For both plots, we observe some similarities. First, using the MW to customise the probabilistic forecasts results in probabilistic forecasts with the narrowest prediction intervals and these prediction intervals appear to noticeably

underestimate the uncertainty. Second, using the EQD as the loss metric for customisation results in the widest prediction intervals, which appear to overestimate the uncertainty. Third, the probabilistic forecasts generated when using the MAQD or UQD are almost identical.

Comparing the two plots, we also observe some differences. There is far less uncertainty in the probabilistic forecasts generated with the TFT as a base forecaster than those generated with the LR as a base forecaster. Furthermore, this uncertainty in the forecasts generated from the TFT is more closely correlated to the fluctuations in the ground truth. The probabilistic forecasts generated when using the LR as the base point forecaster often have strange downward spikes in uncertainty, whilst those from the TFT only spike when the ground truth also spikes.

Calibration Plots To gain more insight into the calibration of the different forecasts generated when using various loss metrics to customise them, we plot exemplary calibration plots on the Electricity data set for the LR in Figure 5.5 and the TFT in Figure 5.6. Both plots demonstrate many similarities. First, the probabilistic forecasts generated when using the MW to customise them show, by far, the worst calibration. Second, for both base point forecasters, using the PL as a loss metric in the loss-customisation results in the second worse calibration. These probabilistic forecasts resulting from the PL slightly overestimate the lower quantiles and underestimate the higher quantiles. Third, the probabilistic forecasts generated when using the remaining four loss metrics in the loss-customisation are all well calibrated. Specifically, those forecasts generated when using the MAQD and CRE as a custom loss metric are almost perfectly calibrated for both base point forecasters.

The only noticeable difference between the two base point forecasters is when considering the probabilistic forecasts generated with the EQD and UQD as the loss metrics. For the TFT these loss metrics still result in probabilistic forecasts that are almost perfectly calibrated, whilst for the LR both metrics lead to probabilistic forecasts that underestimate the lower quantiles and overestimate the upper quantiles.

5.4.3 Effects of Hyperparameter Optimisation

In the third step of our evaluation, we analyse the effects of the HPO. To this means, we first compare how the optimal sampling hyperparameter $\sigma^\star$ changes, depending on the loss metric used in the HPO. Second, we compare multiple evaluation criteria simultaneously to identify if a certain loss metric results in high performance in more than one evaluation criterion at the same time. Finally, we report the computational time of our approach.

Optimal Sampling Hyperparameter To analyse how the optimal sampling hyperparameter changes depending on the loss metric used in the HPO we plot the optimal $\sigma^\star$ for each loss metric and data set in Figure 5.7. Furthermore, we report the optimal sampling hyperparameter for all loss metrics and data sets in Table 5.5. We first observe that for all data sets, using the MW

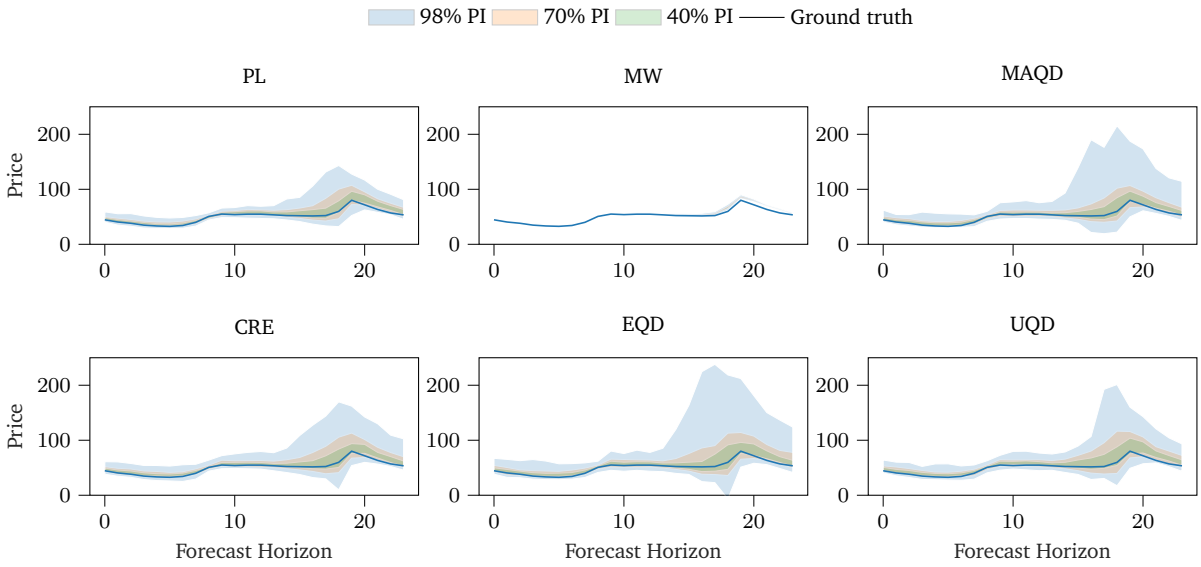


Figure 5.3.: Exemplary 98%, 70%, and 40% prediction intervals on the 11.12.2013 for the Price data set. Probabilistic forecasts are generated by using LR combined with the cINN and customising the probabilistic forecast with various loss metrics.

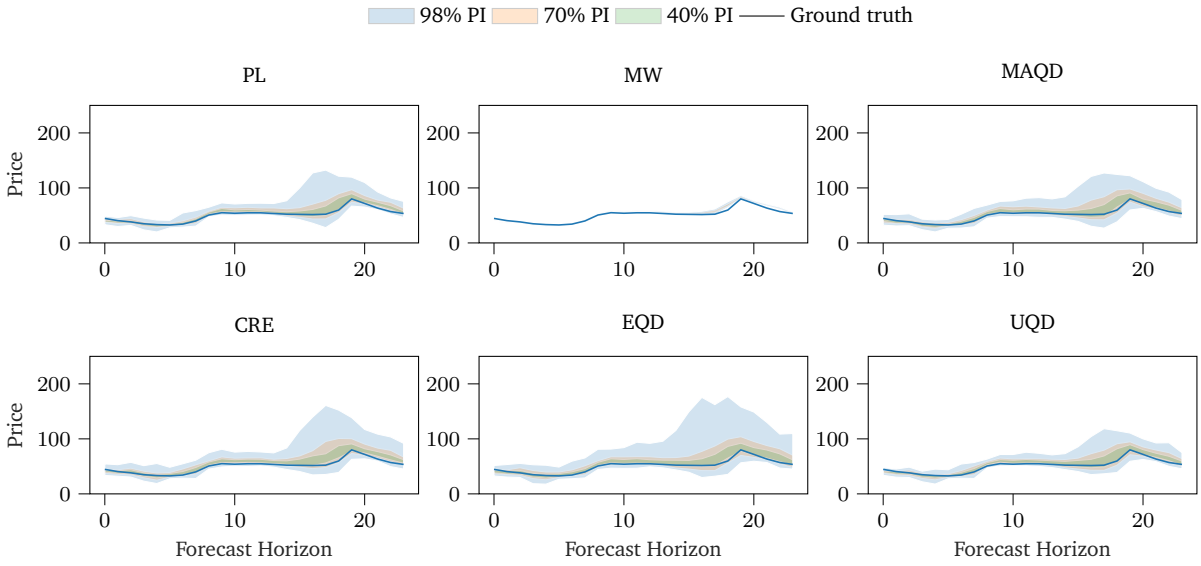


Figure 5.4.: Exemplary 98%, 70%, and 40% prediction intervals on the 11.12.2013 for the Price data set. Probabilistic forecasts are generated by using the TFT combined with the cINN and customising the probabilistic forecast with various loss metrics.

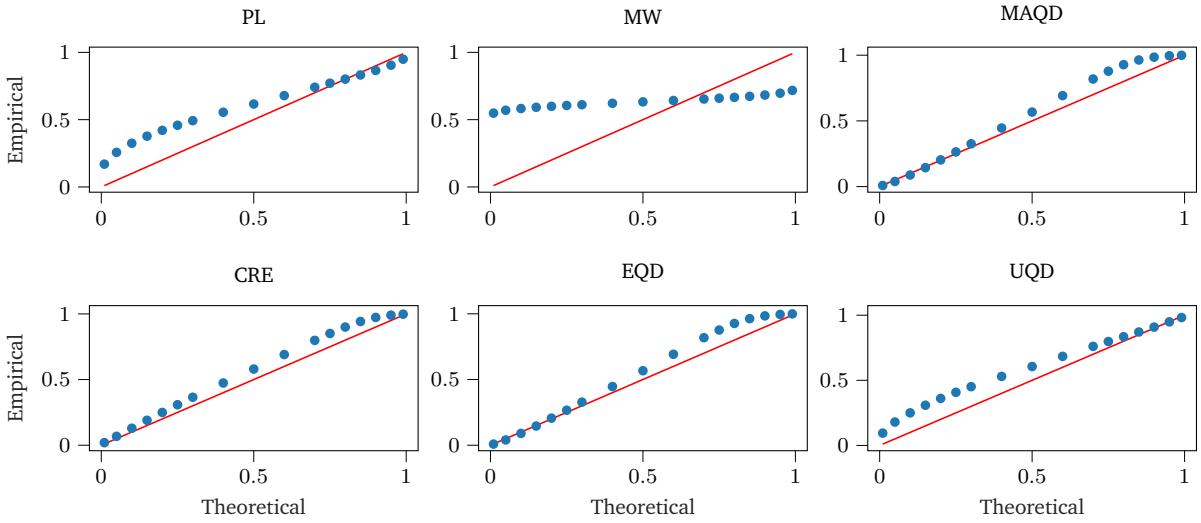


Figure 5.5.: Exemplary calibration plots comparing the theoretical and forecast quantiles on the Electricity data set, with the red diagonal indicating zero deviation. We compare probabilistic forecasts generated by using the LR as a base point forecaster, combining it with the cINN and customising the forecast with different loss metrics.

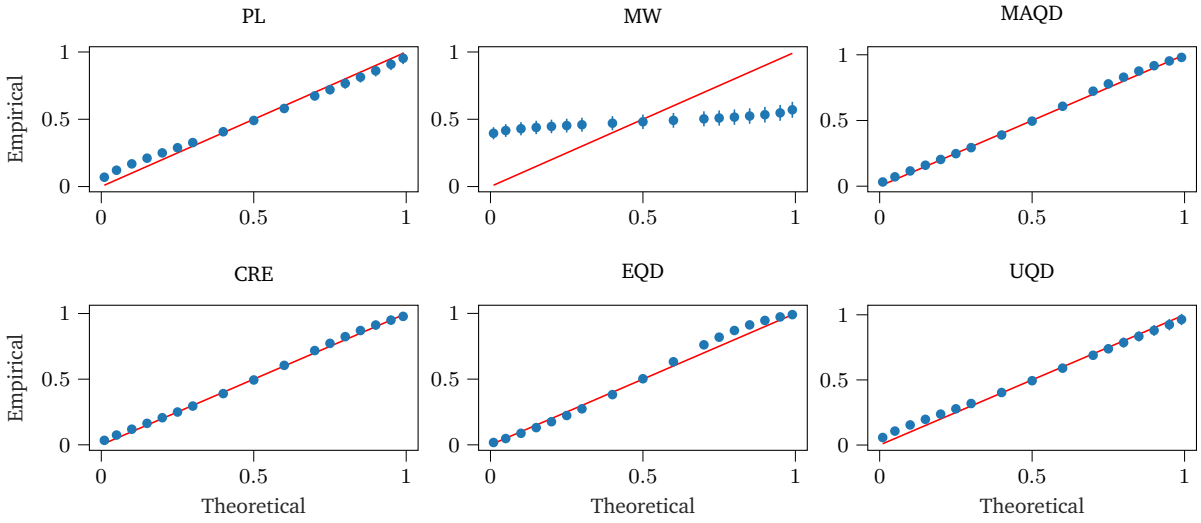


Figure 5.6.: Exemplary calibration plots comparing the theoretical and forecast quantiles on the Electricity data set, with the red diagonal indicating zero deviation. We compare probabilistic forecasts generated by using the TFT as a base point forecaster, combining it with the cINN and customising the forecast with different loss metrics.

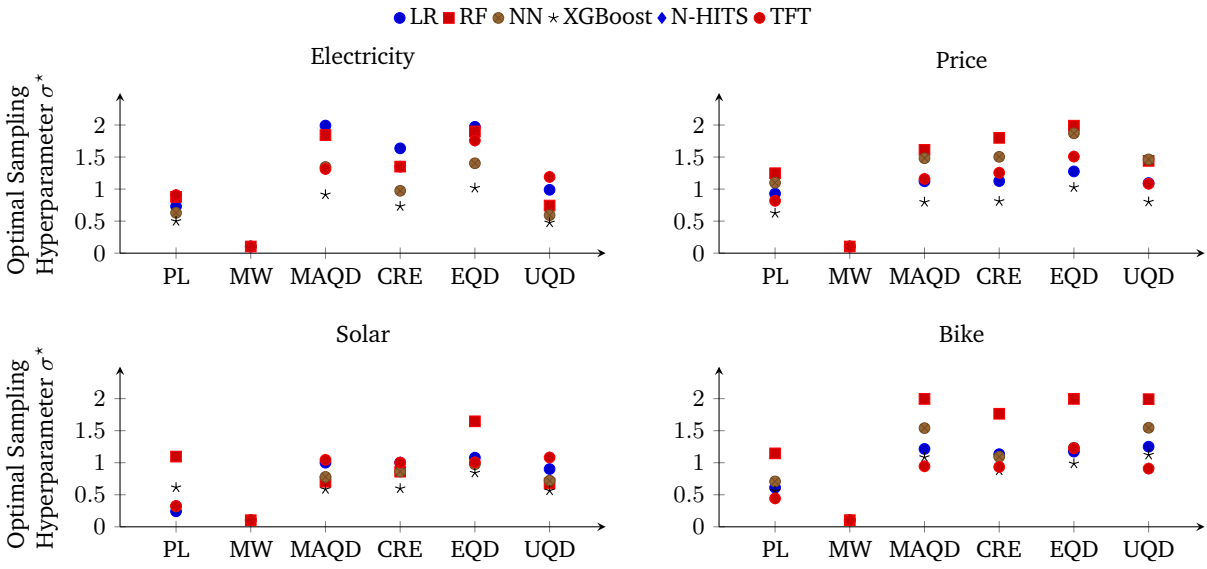


Figure 5.7.: Comparison of the optimal sampling hyperparameter $\sigma^\star$ for the various loss metrics used to customise our probabilistic forecasts and all data sets. These $\sigma^\star$ values are also reported in Table 5.5. [4]

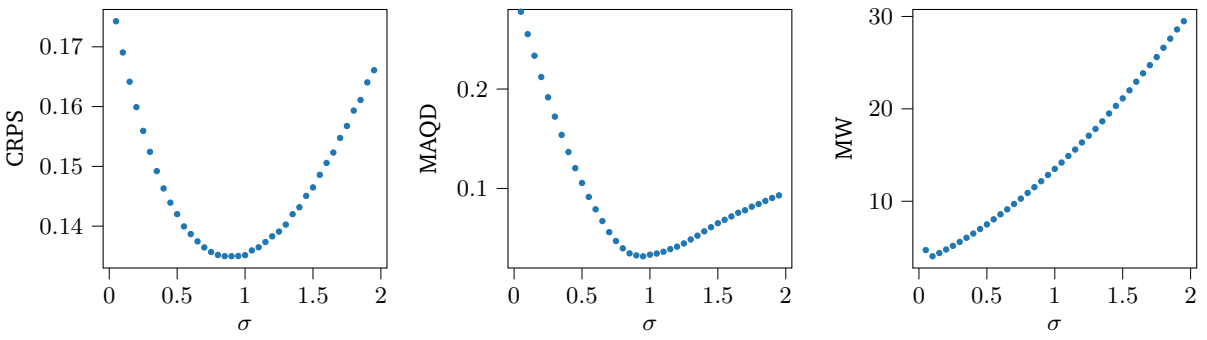


Figure 5.8.: Plots detailing how different values of σ affect the CRPS, MAQD and MW for the TFT on the Price data set. We clearly see that each loss function results in a different $\sigma^\star$.

as the loss metric in the HPO always results in the lowest $\sigma^\star$. Second, the second lowest $\sigma^\star$ is generally identified when using the PL in the HPO, although for certain base point forecasters on certain data sets applying the UQD for loss-customisation can result in a smaller $\sigma^\star$. Third, the remaining loss metrics generally result in a similar $\sigma^\star$ when applied in the HPO. Finally, although the $\sigma^\star$ varies per data set and base point forecaster, the order between the base point forecasters remains similar. For example, the Random Forest (RF) as a base point forecaster often results in the largest $\sigma^\star$, whilst eXtreme Gradient Boosting (XGBoost) often results in the smallest $\sigma^\star$.

To further analyse the effects of the sampling hyperparameter, we compare how the values for the CRPS, MAQD and MW change dependent on a range of sampling hyperparameters on the Price data set with the TFT and visualise these results in Figure 5.8. This observations shows that each loss metric clearly has a different value for $\sigma^\star$, with the optimal for the MW loss metric being the lowest, whilst that for the MAQD is the highest.

Table 5.5.: The optimal sampling hyperparameter $\sigma^\star$ for each base point forecaster when different loss metrics are used in the automated HPO. These $\sigma^\star$ values are also visualised in Figure 5.7.

		Custom Loss Metrics					
Data	**Point Forecaster**	**PL**	**MW**	**MAQD**	**CRE**	**EQD**	**UQD**
Electricity	LR	0.7301	0.1034	1.9915	1.6359	1.9720	0.9890
	RF	0.8781	0.1041	1.8429	1.3507	1.8992	0.7430
	NN	0.6294	0.1041	1.3455	0.9734	1.4026	0.5926
	XGBoost	0.4982	0.1041	0.0913	0.7309	1.017	0.4757
	N-HiTS	0.9063	0.1041	1.3134	1.3500	1.7583	1.1896
	TFT	1.0455	0.1041	1.4646	1.4321	1.8067	1.1519
Price	LR	0.9291	0.1041	1.1215	1.1245	1.2756	1.0942
	RF	1.2472	0.1041	1.6118	1.7998	1.9893	1.4404
	NN	1.0990	0.1041	1.4081	1.5031	1.8703	1.4627
	XGBoost	0.6233	0.1041	0.7965	0.8099	1.0264	0.8003
	N-HiTS	0.8172	0.1034	1.1586	1.2055	1.5076	1.0828
	TFT	1.0268	0.1041	1.4419	1.4161	1.6619	1.3363
Solar	LR	0.2414	0.1027	1.0002	1.0008	1.0788	0.9012
	RF	1.0958	0.1027	0.7029	0.8608	1.6463	0.6766
	NN	0.3175	0.1041	0.7787	0.8625	0.9755	0.7202
	XGBoost	0.6133	0.1027	0.5862	0.5966	0.8414	0.5644
	N-HiTS	0.3026	0.1034	1.0456	1.0027	1.0074	1.0817
	TFT	0.6925	0.1041	1.2253	1.0692	1.2431	1.1618
Bike	LR	0.6116	0.1034	1.2155	1.1338	1.1736	1.2503
	RF	1.1456	0.1034	1.9958	1.7641	1.9965	1.9918
	NN	0.7085	0.1041	1.5401	1.0093	1.2339	1.5458
	XGBoost	0.5584	0.1034	1.0842	0.8762	0.9845	1.1213
	N-HiTS	0.4436	0.1021	0.9448	0.9347	1.2235	0.9085
	TFT	0.4612	0.1027	0.7676	0.7469	1.0459	0.7811

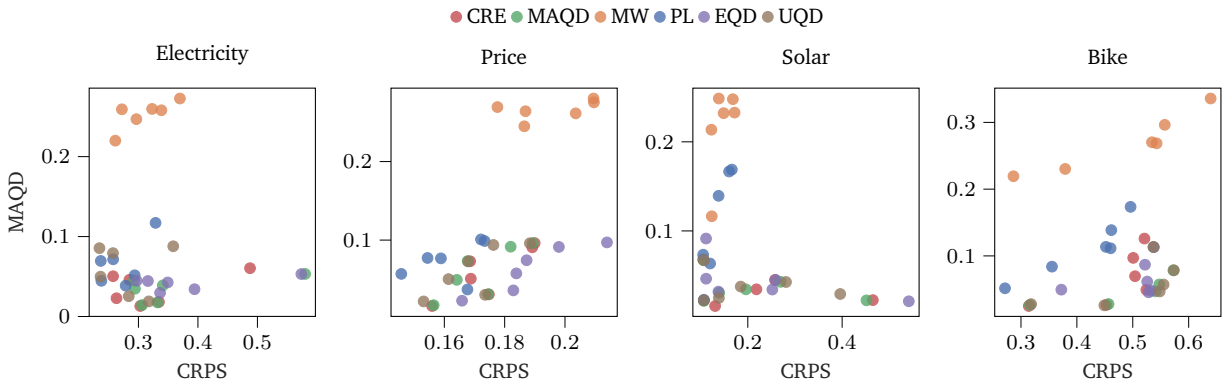


Figure 5.9.: Comparison of the average CRPS and MAQD across three runs for all combinations of base point forecaster and custom loss metric. Note that these combinations are grouped according to the custom loss metric used in the HPO and not differentiated by the base point forecaster applied.

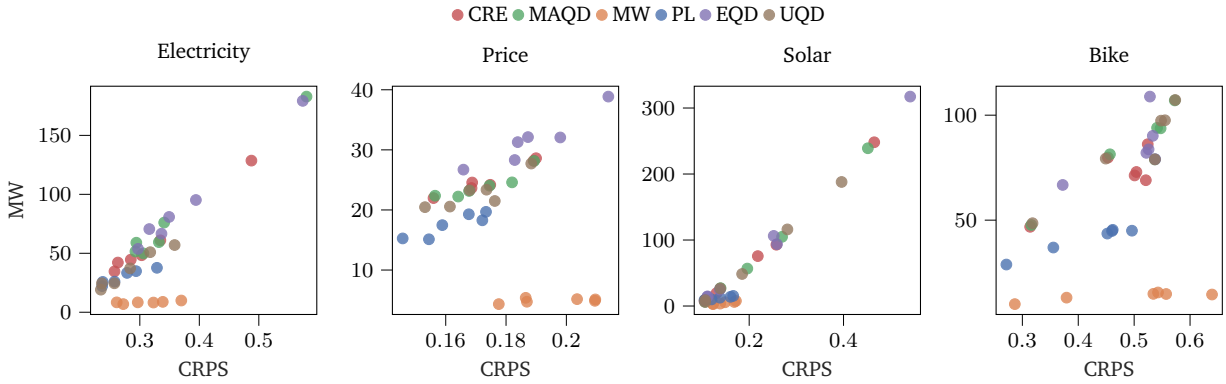


Figure 5.10.: Comparison of the average CRPS and MW across three runs for all combinations of base point forecaster and custom loss metric. Note that these combinations are grouped according to the custom loss metric used in the HPO and not differentiated by the base point forecaster applied.

Comparison of Evaluation Criteria With our approach, we generate loss-customised probabilistic forecasts that are customised according to one loss metric. To investigate whether a specific loss metric results in high performance according to more than one evaluation criterion, we compare multiple criteria simultaneously. Thereby we compare the CRPS and MAQD in Figure 5.9, the CRPS and MW in Figure 5.10, and the MAQD and MW in Figure 5.11.

When comparing the CRPS and MAQD in Figure 5.9, we first observe that using the MW in the HPO never results in the best performance in either metric. Although the resulting CRPSs are not the worst, the MAQDs when using the MW are always worse than the other metrics. Second, no clear loss metric results in optimal performance in both loss metrics on all data sets. However, we observe that generally, forecasts customised using the PL or UQD perform well in both metrics on all data sets. Furthermore, probabilistic forecasts customised with the CRE or the MAQD also achieve reasonable performance in both evaluation metrics on all data sets.

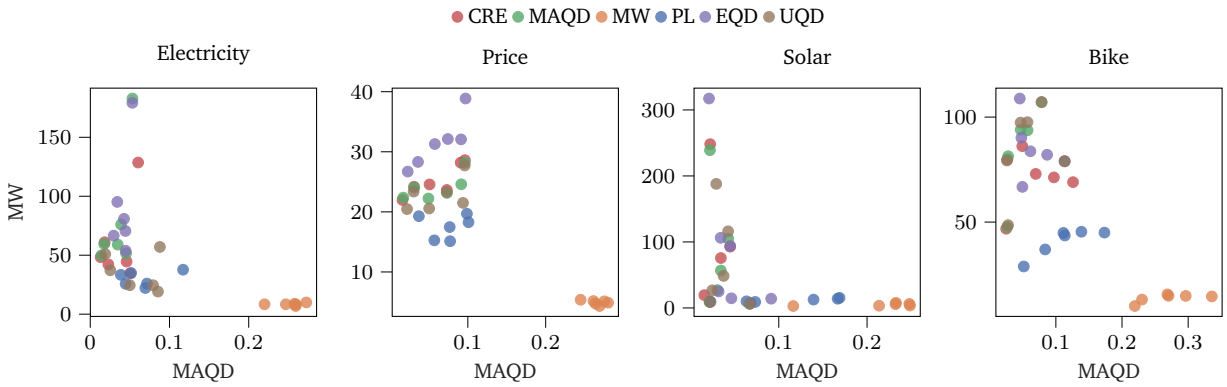


Figure 5.11.: Comparison of the average MAQD and MW across three runs for all combinations of base point forecaster and custom loss metric. Note that these combinations are grouped according to the custom loss metric used in the HPO and not differentiated by the base point forecaster applied.

With regards to the CRPS and MW in Figure 5.10, we note that the probabilistic forecasts customised with the MW always achieve best performance with regards to MW and perform reasonably well with regards to CRPS on the Solar and Electricity data sets. Furthermore, there is no loss metric that achieves good performance on both metrics for all data sets. For the Price and Bike data sets, the PL results in the lowest CRPS and the second lowest MW values, whilst the UQD achieves the same on the Solar and Electricity data sets.

Finally, when comparing the MAQD and MW in Figure 5.11, we observe far more diversity across the data sets. Only on the Solar data set do we observe that using the UQD in the HPO generates probabilistic forecasts that perform well according to MAQD and MW. The distribution of the various probabilistic forecasts also varies across the data sets. For the Electricity data set, a large clump of probabilistic forecasts achieves a low MAQD and a MW of around 50. In the Bike and Solar data sets, we observe a range of probabilistic forecasts with similar MAQD but increasing MW. On the price data set we also observe a cluster of results with a MAQD smaller than one, and MW varying from around 15 to 40.

Computation Time We report the computation time for our approach to generate loss-customised probabilistic forecasts in Table 5.6. For each data set, the cINN is only trained once, with training times varying between $22.42\,\mathrm{s}$ and $32.9\,\mathrm{s}$. Furthermore, for each data set, the base point forecasters are also trained once. The training times for the base point forecasters are highly individual, ranging from $0.1\,\mathrm{s}$ for the LR on the Solar and Bike data sets, to $1044\,\mathrm{s}$ for the TFT on the Bike data set. Given the trained cINN and base point forecasters, we focus on the time to customise the forecast, i.e. the time taken to generate and evaluate a single loss-customised forecast.[3] This time to customise the forecast varies only depending on the data set and is almost identical for each of the considered base point forecasters. This time is the lowest on the Bike data set, with an

[3] As described in Section 5.3.2, we use a tuner timeout of $120\,\mathrm{s}$ for the automated HPO and no additional stopping criteria. Therefore, the total time taken for the automated HPO is always approximately $120\,\mathrm{s}$ and, thus, not interesting to report.

average time of $8.87\,\mathrm{s}$ across the point forecasters, and the highest on the Electricity data set, with an average time of $13.15\,\mathrm{s}$ across all point forecasters. In addition and for comparative purposes, we report the training time of the direct probabilistic benchmarks from the previous chapter (see Section 4.3) in Table 5.7. Generally, the Nearest Neighbour Quantile Filter (NNQF) requires less computation time than DeepAR and the Quantile Regression Neural Network (QRNN), although training times vary across all data sets.

5.5 Discussion

In this section, we first present some key insights from our evaluation before discussing the limitations and benefits of our proposed approach for creating loss-customised probabilistic forecasts.

Key Insights Based on our evaluation, we gain five key insights. First, we observe that the selected loss metric applied in the automated HPO does effectively customise the resulting probabilistic forecast. These loss-customised forecasts are based on noticeably different optimal sampling hyperparameters and differ visually with respect to the resulting Prediction Intervals (PIs) and the calibration plots. Furthermore, their performance in the four considered evaluation criteria varies noticeably depending on the loss metric used for the automated HPO.

Second, we observe expected effects depending on which loss metric we use. The probabilistic forecasts customised with the PL, for example, always perform best with regards to CRPS. Since the CRPS can be considered as a continuous extension of the PL this result is not surprising. Furthermore, the PL results in probabilistic forecasts that perform mid-range in terms of the calibration and sharpness measures. This is also logical since the PL implicitly considers both calibration and sharpness and is, therefore, likely to result in loss-customised probabilistic forecasts that are reasonable in both metrics. On the other hand, the calibration-based metrics, i.e. the MAQD, CRE, EQD and UQD, always result in the probabilistic forecast with the best calibration and worse sharpness. Importantly, no single calibration-based metric outperforms the others, and the best-performing metric depends on the base point forecaster and data set. However, this is not surprising since all metrics are similar and differ only in how exactly they penalise miss-calibration in forecasts. Therefore, it makes sense that the best-performing metric depends on the data considered and how the point forecast predicts this data, i.e. a data set with many extreme values may be better calibrated when the EQD or UQD is applied since these loss-metrics weight extreme quantiles higher. Finally, the probabilistic forecasts generated when using the MW in the HPO are always the sharpest, resulting in the lowest MW scores and the smallest $\mathrm{nMPI}(\beta)$s, but poor calibration. Furthermore, the MW values for the calibration-based loss metrics are generally the highest. This is also logical since the calibration-based loss metrics often result in wide prediction intervals to account for extreme regions of uncertainty, which adversely affect the MW values.

Third, we also observe similar results between certain loss metrics. More specifically, the forecasts customised with calibration metrics are all well-calibrated and have similarly wide prediction intervals. Furthermore, the forecasts generated when using the MAQD or UQD in the HPO are almost identical, suggesting that the symmetry in the quantiles does allow us to achieve a similar loss-customisation when only considering the upper half.

Fourth, we observe that none of the considered loss metrics can consistently achieve the best performance in more than one evaluation criterion. When comparing different evaluation criteria, none of the considered loss metrics resulted in loss-customised probabilistic forecasts that consistently performed highly in more than one of the considered metrics. This further highlights that the desired properties of probabilistic forecasts are not always complementary, and it is still a significant challenge to generate probabilistic forecasts that perform well with regard to multiple criteria.

Finally, for complex base point forecasters, we observe that our approach is computationally cheap when compared to complete retraining. The time taken to generate a loss-customised forecast is approximately $10\,\mathrm{s}$, which is noticeably less than the training time of all considered base forecasters, apart from LR. Furthermore, this is noticeably less than the training time for each of the direct probabilistic forecasting benchmarks from Chapter 4.[4]

Limitations Our approach has a few limitations. First, our approach trains multiple base point forecasters and then uses automated HPO to select an optimal sampling hyperparameter based on a custom loss metric. As a result, the base point forecaster must be manually selected. However, ideally, we require a fully automated forecasting process. Therefore, to enable the application of our approach in an automated setting, an optimal base point forecaster, its hyperparameters, and the appropriate sampling hyperparameter should all be optimised based on a given probabilistic loss metric in a fully automated manner. Second, the automated HPO in our approach is currently only terminated by a tuner timeout of $120\,\mathrm{s}$. However, it may be possible to achieve similar results in a fraction of the time. Therefore, further convergence criteria should be considered to further reduce computation time.

Benefits We consider several aspects of our approach particularly beneficial. First, our approach is capable of creating loss-customised probabilistic forecasts based on an arbitrary custom loss metric. This arbitrary loss metric is important because probabilistic forecasts are often inputs for further downstream applications such as optimisation tasks. Therefore, if a custom loss metric can be designed based on the requirements of this downstream application, our approach can generate loss-customised probabilistic forecasts that are ideal for this application. For example, in a safety-critical application that requires an overestimation of uncertainty, a loss-customised probabilistic forecast based on a loss metric similar to the EQD would be far more beneficial than existing probabilistic forecasts.

[4] Since the considered benchmarks do not allow a custom loss metric to be used in training, this comparison is based on the assumption that such retraining is possible and that a similar training time can be expected.

Second, our approach can work with an arbitrary base point forecasting model thanks to the applied forecasting approach. Therefore, if a point forecasting model has already been designed to perform particularly well in a specific application, our approach can be applied to generate multiple loss-customised probabilistic forecasts without modifying the existing model. Furthermore, if a custom loss metric also exists for this application, our approach can be directly applied to generate loss-customised probabilistic forecasts without changing the existing setup.

Finally, our approach generates loss-customised probabilistic forecasts without retraining the applied cINN or the base point forecasts. Particularly for complex base point forecasters, the time required to generate a new loss-customised probabilistic forecast is negligible compared to the retraining time. As a result, it is computationally cheap to generate several forecasts that differ in the loss function used for the automated HPO, which could be the basis for an ensemble.

5.6 Conclusion

To generate probabilistic forecasts with customised properties, we extend our approach from Chapter 4 and apply automated HPO to generate loss-customised probabilistic forecasts. Our approach includes uncertainty by first mapping a point forecast from an unknown distribution to a representation in a known and tractable distribution with a Conditional Invertible Neural Network (cINN). We then generate a probabilistic forecast by optimising a sampling hyperparameter in an automated manner which includes uncertainty from the neighbourhood of this representation into the point forecast. This automated HPO is based on custom loss metrics, which alter the characteristics of the resulting probabilistic forecast. Thus, our approach is capable of generating loss-customised probabilistic forecasts without retraining either the applied base point forecaster or the cINN.

We evaluate our approach by generating loss-customised probabilistic forecasts on four data sets with six different loss metrics. With a quantitative and qualitative comparison of these probabilistic forecasts, we show that the characteristics vary noticeably depending on the loss metric used in the HPO optimisation. Furthermore, we show that our approach is computationally inexpensive and that none of the considered loss metrics are capable of generating probabilistic forecasts that perform highly in more than one evaluation criterion.

Given our promising results, future work should first evaluate our approach on further data sets. Second, our approach to generate loss-customised probabilistic forecasts should be employed for exemplary applications by first designing custom loss metrics for these applications and then applying them in our approach. Third, the possibilities of applying an ensemble of forecasts generated from our approach should be investigated, including weighting and combining the forecasts to perform better across multiple evaluation metrics. Finally, it would be interesting to fully automate the entire probabilistic forecasting process by also selecting the optimal base point forecaster and the associated hyperparameters in an extensive automated probabilistic forecasting framework.

Table 5.6.: The average computation time across three runs evaluated in seconds for our approach to generate loss-customised probabilistic forecasts. The cINN and each base point forecaster only need to be trained once for each data set (Train cINN and Train Forecaster). With a tuner timeout of 120 s, our automated HPO selects the optimal $\sigma^\star$, where each loss-customised forecast is generated and evaluated (Customise Forecast). [4]

Point Forecaster	Electricity			Price			Solar			Bike		
	Train cINN	Train Forecaster	Customise Forecast	Train cINN	Train Forecaster	Customise Forecast	Train cINN	Train Forecaster	Customise Forecast	Train cINN	Train Forecaster	Customise Forecast
LR		0.11	13.27		0.13	12.99		0.10	9.62		0.10	8.91
RF		41.71	13.16		91.82	12.81		46.28	9.23		51.80	8.85
NN	32.90	70.82	13.20	31.10	72.60	12.98	24.33	55.53	9.56	22.41	51.40	8.89
XGBoost		159.55	13.10		332.13	12.94		200.78	9.59		161.89	8.94
N-HITS		57.83	13.13		64.26	12.45		72.81	9.50		108.65	8.88
TFT		534.43	13.04		426.92	12.54		615.12	9.28		1044.00	8.74

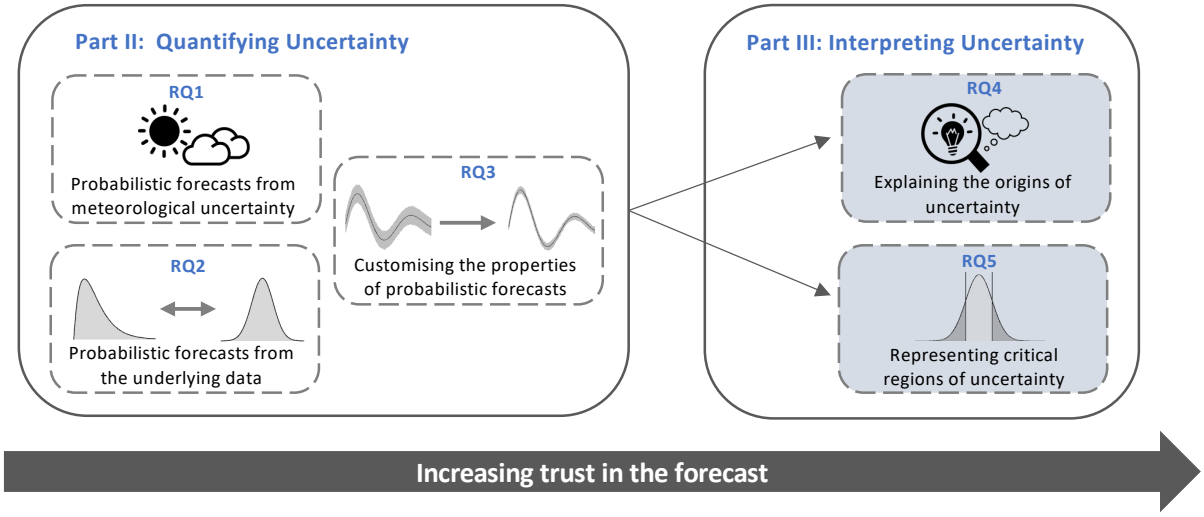
Table 5.7.: The average training time across three runs evaluated in seconds for each of the considered direct probabilistic benchmarks from Section 4.3 on all data sets. [4]

Benchmark	Electricity	Price	Solar	Bike
DeepAR	501.09	533.01	458.97	396.53
QRNN	573.61	503.47	1090.32	686.38
NNQF	496.48	120.37	250.26	116.96

Part III

Interpreting Uncertainty

Overview of Part III



Part III of the dissertation deals with interpreting the uncertainty that was previously quantified with probabilistic time series forecasts, for example, with methods from Part II. We first consider **RQ4** in Chapter 6 by suggesting an approach that uses existing Explainable Artificial Intelligence (XAI) methods to generate explanations for the origins of the uncertainty in a probabilistic forecast. Our approach separates the deterministic and uncertain components of the probabilistic forecast in the network architecture and, thus, allows for separate explanations to be generated for each part. Our results show that the explanations appear plausible, especially for synthetic data, and deliver important insights which could be particularly useful for further model development.

We further address **RQ5** in Chapter 7 by considering critical regions of uncertainty for mobility applications. Focusing on the use case of parking duration predictions for electric vehicle smart charging applications, we highlight how forecasts that overestimate the parking duration, i.e. the electric vehicle leaves earlier than expected, are particularly crucial. Such forecasts could lead to an undercharged vehicle that is unable to reach the final destination. To address this problem, we present multiple representations of the uncertainty which can be applied to mitigate this critical error, even when large amounts of uncertainty are present.

6

Explaining the Origins of Uncertainty in Probabilistic Forecasts

In the previous chapters, we presented multiple methods to generate probabilistic forecasts and thus quantify the uncertainty associated with the time series forecasts. The majority of these approaches, along with almost all state-of-the-art time series forecasts, are based on machine learning and are therefore considered as black box models [201]. Despite the high forecast quality generated by such models, it is often unclear which input factors are influencing the model to generate the resulting forecast [201]. This lack of transparency can make it difficult to trust the forecast and rely on it when making important decisions [202], [203], especially as time series forecasting models are increasingly more complicated with millions of parameters [152], [204], [205]. As a result, to increase trust in such probabilistic forecasts, it is important to increase transparency by explaining which factors influenced the model when generating the forecast [25], [29].

Creating such explanations is the focus of the relatively new research field XAI [206]. The main aim of XAI is to develop transparent machine learning models, i.e. machine learning models which enable users to better understand the results generated by the model [207]. Although such machine learning models ideally offer intuitive explanations and are naturally *interpretable*, XAI also focuses on explaining machine learning models that are not naturally interpretable with so-called post-hoc explanation methods [208]. There are a host of such post-hoc explanation methods, which are often based on perturbing the input features and documenting the effects [46], [209], [210] or propagating the gradients of the applied network model back to the input [47], [48], [211], [212]. Although these methods were often designed for computer vision tasks, they generally only require that the gradients of the model can be propagated back to the inputs [208]. As a result, such post-hoc XAI methods can technically be applied to probabilistic time series forecasting models as long as they fulfil this simple requirement.

Despite the lack of technical barrier, XAI methods have up until now only sparingly been applied to time series tasks and almost never to time series forecasting [49]. Although there is a recent shift towards creating interpretable machine learning models for time series forecasting [40], [213], [214], only Li *et al.* [38] focus on generating probabilistic forecasts and the interpretations provided are in the form of simple feature importance. Furthermore, many state-of-the-art machine learning approaches for probabilistic time series forecasting already exist and are not natively interpretable [34]. To continue using such models in the future, where the importance of interpretable forecasts grows [29], it is thus necessary to apply post-hoc explanation methods

to explain the origins of uncertainty in such models. However, to the best of our knowledge, no existing work has focused on using XAI methods to explain this uncertainty.

Therefore, in the present chapter, we propose a methodology to apply existing XAI methods to explain the origins of uncertainty in a probabilistic forecast. By specifically separating the probabilistic forecast via the neural network architecture into a deterministic component and a component that represents the uncertainty, we enable post-hoc XAI methods to determine the effects of all input features on each of these components separately. We evaluate our approach with a simple neural network by comparing different XAI methods on a synthetic data set and also analysing the usefulness of explanations on four real-world data sets. As a result, the contribution of this chapter is twofold: (1) we demonstrate the viability of combining XAI and probabilistic forecasts to increase trust in the forecast, and (2) we highlight how these explanations deliver valuable insights which may help to improve model design.

The rest of the present chapter is structured as follows. In Section 6.1, we present related work and highlight the identified research gap. We then introduce our methodology in Section 6.2 before explaining the experimental setting used to evaluate our approach in Section 6.3. In Section 6.4, we evaluate our approach and in Section 6.5, we discuss the results and the main insights gained. Finally, we conclude and present future work in Section 6.6.

6.1 Related Work

Whilst the application of XAI on time series is generally limited, the limited existing work mostly focuses on time series classification and not forecasting [49], [215], [216]. In this field, Mochaourab *et al.* [217] and Kenny *et al.* [218] both apply post-hoc explainability methods on time series classification models, whilst Turbé *et al.* [219] also consider time series classification and perform a systematic evaluation of various post-hoc explainability methods. Furthermore, explainability methods are applied to classification in the health domain by Di Martino and Delmastro [220]. These approaches all apply existing post-hoc explanations that are not specifically developed for time series and, therefore, do not consider the temporal dynamics of the time series. To overcome these issues, both Tonekaboni *et al.* [221] and Munir *et al.* [222] propose post-hoc explanation methods that can better capture the temporal dynamics of a model in a time series classification tasks. However, both of these methods are only applied for time series classification and not time series forecasting.

When XAI is applied to time series forecasts, it is almost always via specific models that are designed to be interpretable [49]. Such models rely on model-specific mechanisms to generate explanations and, as a result, cannot be applied post-hoc to explain the output of an existing forecasting model. Such models for point forecasts use, for example, a network *stack* that separates the seasonal and trend component of the forecast within the model [213], multilevel wavelet decomposition [214], or series saliency [223]. Furthermore, Hertel *et al.* [224] explore using the attention scores from transformer models to explain which regions of the history

input are important for transformer-based load time series forecasts. For probabilistic forecasts, interpretable models exist that make use of attention [40], interpret forecasting as a classification task [39], or use relevance scores from an ensemble of deep neural networks [42].

Hardly any work applies post-hoc explainability methods to time series forecasting. Although post-hoc explanations have been used to determine feature importance in time series predictions, for example, by Kruse *et al.* [225] and Trebbien *et al.* [226], these explanations are static and consider the total feature importance and not temporally dependent feature importance. Additionally, Choi *et al.* [227] evaluate their point forecasting architecture with post-hoc explainability methods, Çelik *et al.* [228] apply post-hoc explanations to financial data, Wang *et al.* [229] investigate post-hoc explanations methods for tunnel boring machine time series data, and Barredo Arrieta *et al.* [230] explore the explainability of deep echo state networks. However, in each of these papers, only point forecasts are considered, and there is no attempt made to explain the origins of the uncertainty.

Therefore, we identify a clear lack of research that applies post-hoc explanation methods to explain the origins of the uncertainty in a probabilistic forecast. As a result, in the present chapter we present a methodology that enables existing XAI methods to be applied to explain the origins of uncertainty in a probabilistic forecast.

6.2 Explaining the Origins of Uncertainty

An overview of our approach to explain the origins of uncertainty in time series forecasts with existing XAI methods is presented in Figure 6.1. First, the input data is preprocessed and used to train a gradient-based probabilistic forecasting model. This trained model and the associated inputs are then combined in a post-hoc explainability method [206] to generate two explanations: an explanation for the mean forecast and an explanation for the uncertainty associated with this forecast. To enable this approach, we have to separate both components of the forecast within the network architecture. Therefore, this section briefly highlights the approach used to generate such probabilistic forecasts before explaining the idea between the post-hoc explanation methods used, so-called attribution-based explanation methods [206]. Finally, we outline how both aspects can be combined to explain the origins of uncertainty.

6.2.1 Probabilistic Forecasting Approach

To explain the origins of uncertainty, we first need to isolate the uncertainty in the probabilistic forecast. We achieve this isolation through a specific neural network design scheme, schematically shown in Figure 6.2. As inputs, this schematic network considers historical information $\mathbf{y_{t-P}}$ and forecasts for M exogenous features $\hat{\mathbf{x}}_{\mathbf{m,t+H}}$, where $m \in \{1, 2, \ldots, M\}$. Although not explicitly shown in the above figure, it is possible that the network only considers historical information, i.e. $M = 0$, or that each of these inputs is processed by a separate encoder network before entering

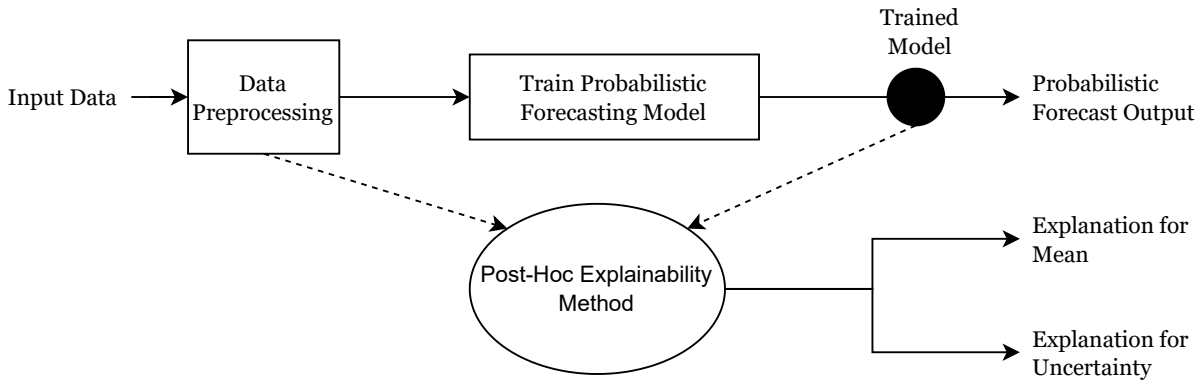


Figure 6.1.: An Overview of the approach to explain the origins of uncertainty in probabilistic forecasts. After preprocessing, the selected input features and history data are used to train the probabilistic forecasting model. This trained model is then capable of generating probabilistic forecasts. The trained model and input data are combined with a post-hoc explainability method to generate explanations for the mean component and uncertainty component of the forecast.

the main network. The main network consists of hidden layers that consider all of the input features. This main network is also not limited to a simple feed-forward architecture and can, therefore, include more complicated layers, such as long short-term memory (LSTM), or even self-attention via transformers. The important aspect of the proposed schematic architecture is the output. To isolate the deterministic and uncertain components of the forecast, we create two separate output networks. The first of these networks generates a mean forecast $\hat{\boldsymbol{\mu}}_{\mathbf{t+H}}$, whilst the second network generates a variance forecast $\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_{\mathbf{t+H}}$. These two outputs can be combined in a parametric distribution, e.g. a Gaussian distribution, to generate distributional probabilistic forecasts, see Morales *et al.* [45]. Importantly, the architecture of these output networks is also not limited and can, again, include diverse layers.

Since we consider H-step ahead forecasts, the final probabilistic forecast is also H-dimensional. To achieve this, each of the time steps within the forecast horizon $t+1, \cdots, t+H$ is realised as a single output. More specifically, each of the two output networks generates a H-dimensional output. As a result, the final output dimension of the neural network is $2 \times H$. Each of the inputs is also multi-dimensional. Each forecast exogenous feature is H dimensional, whilst the history input is P dimensional. As a result, the neural network has an input dimension of $1 \times (M \cdot H + P)$.

Such a network can be trained to generate probabilistic forecasts by assuming a parametric distribution and applying the maximum likelihood method to maximise the likelihood of the data given the assumed parametric distribution [231]. Therefore, the schematic architecture can be applied to any parametric distribution which allows approximation via the maximum likelihood method. We detail the training and concrete architecture of the network used to evaluate our approach in Section 6.3.

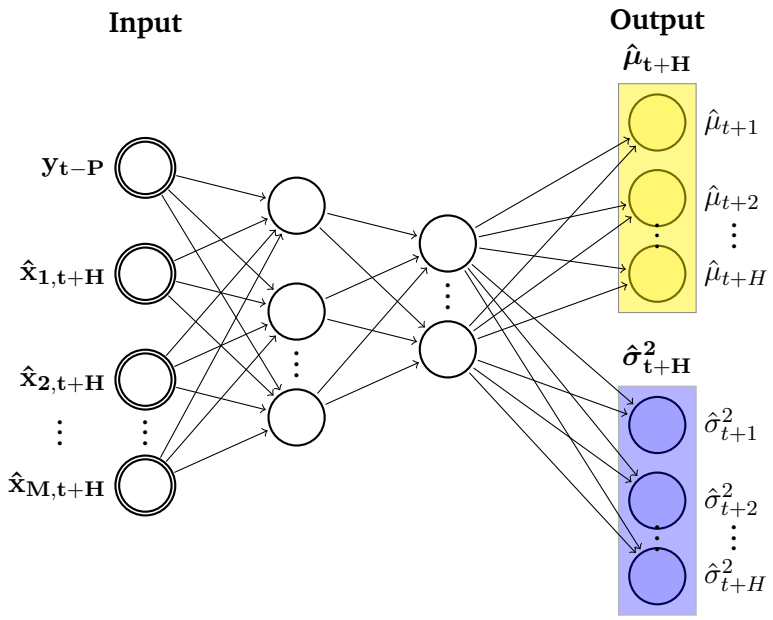


Figure 6.2.: A schematic overview of the neural network architecture used to generate probabilistic forecasts. The network uses historical information $\mathbf{y_{t-P}}$ and M exogenous feature forecasts $\hat{\mathbf{x}}_{\mathbf{m,t+H}}$ to generate probabilistic forecasts for horizon H. These inputs are vectors as they contain values for multiple points in time, which we indicate with double circles for the input neurons. The output of the network separates the deterministic and uncertain components of the probabilistic forecast by forecasting the mean and variance of the forecast as separate outputs. Furthermore, each time step in the forecast horizon is a separate output.

6.2.2 Attribution-Based Explanation Methods

We use post-hoc attribution-based XAI methods to explain the origins of uncertainty in the probabilistic forecasting model. Such attribution methods are designed to produce explanations by assigning a scalar attribution value to each dimension of each input feature of the considered machine learning model for a specific sample [232]. To explain this concept, we consider a simplified setting with a trained neural network that only considers a single feature vector as input and generates a forecast consisting of only a single output vector. Specifically, this simplified trained neural network g considers the single input vector $\mathbf{x} = x_1, x_2, \ldots, x_I$ and generates the output $\mathbf{y} = y_1, y_2, \ldots, y_H$, i.e.

$$\mathbf{y} = g(\mathbf{x}). \tag{6.1}$$

Given such a network, an attribution-based explanation results in an attribution matrix A for a single output sample, i.e.

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1H} \\ a_{21} & a_{22} & \ldots & a_{2H} \\ \vdots & \vdots & \vdots & \vdots \\ a_{I1} & a_{M2} & \ldots & a_{IH} \end{bmatrix}, \tag{6.2}$$

where a_{ij} is the attribution for the $i-th$ dimension of the input explaining the $j-th$ dimension of the networks output [233]. For time series, each of these dimensions represents a different

point in time and therefore, such an attribution matrix can be used to explain how different points in time in a given input feature affect the output.

There are multiple classes of attribution XAI methods, with the most common of these being gradient-based attribution methods [232] and perturbation-based attribution methods [233]. Gradient-based methods rely on calculating gradients, specifically the gradients of the model's output concerning its input features, providing explanations by quantifying how small changes in input features affect predictions [232]. In contrast, perturbation-based methods involve introducing controlled perturbations to the input data, observing their impact on the model's predictions, and approximating feature importance by comparing the model's behaviour on perturbed inputs to the original input [233]. Although not dependent on backpropagation, such perturbation methods are often more computationally expensive. We consider both approaches in the present chapter and explain the details of the attribution-based XAI methods applied in Section 6.3.

6.2.3 Applying Attributions to Probabilistic Forecasts

After separating the deterministic and uncertainty components of the probabilistic forecast within the neural network architecture, we can apply attribution-based XAI methods to explain the origins of uncertainty. More specifically, we can consider each of these outputs separately and calculate the attribution matrices for each of the input features. In this means, we obtain explanations for both the mean forecast, i.e. the deterministic component of the forecast, and the variance forecast, i.e. the uncertainty.

Importantly, since we are now considering multiple input vectors and two output vectors, the attribution matrix is not as simple as defined above. Specifically, to explain the effects of a single H-dimensional exogenous forecast feature on the uncertainty, we must consider how each of these H dimensions affects the H outputs in the forecast horizon. More specifically, for each of the two output vectors $\hat{\boldsymbol{\mu}}_{\mathbf{t+H}}$ and $\hat{\boldsymbol{\sigma}}^{\mathbf{2}}_{\mathbf{t+H}}$, we obtain an attribution matrix A as above for each input feature into the neural network. This structure results in a $P \times H$ attribution matrix for both the mean and variance prediction when explaining the history input. Furthermore, we obtain a $H \times H$ attribution matrix for each combination of output (mean or variance) and individual input feature. If, for example, we were to consider two exogenous inputs as well as history inputs, we would obtain the following attribution matrices: (1) a $P \times H$ attribution matrix explaining how the history input affects the mean forecast, (2) a $P \times H$ attribution matrix explaining how the history input affects the variance forecast, (3) two $H \times H$ attribution matrices, explaining how each of the exogenous inputs affects the mean forecast, and (4) two $H \times H$ attribution matrices explaining how each of the exogenous inputs affects the variance forecast. Although this structure poses no technical difficulties, it is important to understand the interplay between dimensions being considered.

6.3 Experimental Setting

To evaluate our approach, we apply multiple attribution-based XAI methods on probabilistic forecasts for synthetic and real data.[1] In this section we detail the applied experimental setup. First, we describe the data used before introducing the attribution-based XAI methods applied. We then present the evaluation metrics before finally highlighting the implementation of the probabilistic forecasting models applied.

6.3.1 Data

In this section, we briefly introduce the synthetic data set and the four real-world data sets used in our evaluation. For all data sets, we normalise each feature and target independently and use the first 80% of the data for training and the last 20% for testing.

Synthetic Data To compare different attribution-based XAI methods and consider exemplary explanations, we use a synthetic data set. The synthetic data is simply a sine wave with an additive noise component, i.e. for time t

$$\text{Synthetic}_t = \mathcal{A}_{\text{base}} \cdot \sin(2 \cdot \pi \cdot \omega \cdot t) + \mathcal{A}_{\text{noise}} \cdot \mathcal{N}(0, \sigma^2_{\text{noise}}), \tag{6.3}$$

where $\mathcal{A}_{\text{base}}$ is the base amplitude of the time series, ω the frequency of the sine wave, $\mathcal{A}_{\text{noise}}$ the amplitude of the additive noise and σ^2_{noise} the variance of this additive noise. In the present chapter we create synthetic data with $\mathcal{A}_{\text{base}} = 5$, $\mathcal{A}_{\text{noise}} = 1$, and $\sigma^2_{\text{noise}} = 1$. Furthermore, we consider two different frequencies, $\omega = 0.05$ and $\omega = 0.1$, and generate a time series with a length of 10 thousand.

Real Data We consider four real-world data sets in the evaluation, two of which only contain history inputs and two that also include forecast exogenous features. The first two data sets are taken from the Western European Power Consumption (ENTSO-E)[2] database. These two data sets *Sweden* and *Germany* consist of total electricity consumption for Sweden and Germany, collected by the ENTSO-E between January 2015 and July 2020. The data is originally at a temporal resolution of $15\,\text{min}$, but we resample to a $1\,\text{h}$ resolution for our evaluation. Both the Sweden and Germany load data sets only contain historical information and no forecast exogenous features. To also include data sets with exogenous forecast features, we consider the *Price* and *Solar* data from Global Energy Forecasting Competition 2014 (GEFCom2014) [162], originally introduced in Section 4.3. For the Price data set, we forecast the *Zonal Price*, using the *Total Zonal Load* and *Total Load* as exogenous forecast features. For the Solar data set, we forecast the Solar *Power*

[1] Code to replicate our results is available via GitHub: `https://github.com/kalebphipps/Explaining-the-origins-of-uncertainty`.

[2] `https://www.kaggle.com/datasets/francoisraucent/western-europe-power-consumption`.

using the *Total Column Cloud Liquid Water (TCLW)*, the *Total Cloud Cover (TCC)*, and *Surface Solar Radiation Downwards (SSRD)* as exogenous forecast features.

6.3.2 Applied Attribution Methods

We consider five different attribution-based XAI methods in our evaluation, two of these are gradient-based and three perturbation-based. We implement all attribution-based explanation methods using the Python package Captum [234]. To clearly introduce each of these applied attribution approaches, we require a slightly simplified notation and, therefore, define the combination of all input features for the neural network as

$$\mathbf{F} = [\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_{M+1}}], \tag{6.4}$$

whereby the individual elements of this input matrix $\mathbf{f_l}$ with $l = 1, \ldots, M+1$ correspond to the input features discussed above, i.e. the history data and the forecast exogenous features. Furthermore, we define the single output neuron whose output is being explained as $\mathcal{O}_{\text{target}}$. Given these definitions, we explain each of the attribution-based methods in more detail.

Integrated Gradients The first attribution method we consider is Integrated Gradients (IG), a gradient-based approach [211]. The idea of IG is rather intuitive. Starting with a simple baseline, one gradually changes this input to match the actual input in the model. At each step along this transition from the baseline input to the actual input, IG computes the contribution of each input feature by integrating the gradients of the model's output with respect to this input feature [211]. In the present chapter, we consider a vector of zeros as the baseline vector for all inputs. Therefore, we define IG for feature l as

$$\text{IG}_l(\mathbf{F}) = (\mathbf{f}_l - \mathbf{f'}_l) \cdot \int_{\kappa=0}^{1} \frac{\partial \mathcal{O}_{\text{target}}(\mathbf{F'} + \kappa \cdot (\mathbf{F} - \mathbf{F'}))}{\mathbf{f}_l} \text{d}\kappa, \tag{6.5}$$

where $\mathbf{F}$ is the combined input into the neural network model, $\mathbf{F'}$ the combined baseline input, $\mathbf{f}_l$ and $\mathbf{f'}_l$ the single feature input and baseline single feature input respectively, and κ the integration variable used to integrate over the transitional path between the baseline and the actual input.[3] Therefore, starting with a baseline of zero vectors, we gradually adapt this input to match the actual input, integrating at each step during this transition. This integral is approximated via a Riemann sum in the Captum implementation [234].

Saliency The second attribution explainability method we use is the gradient-based Saliency method proposed by Simonyan *et al.* [212]. Saliency is simpler than IG and involves calculating the partial derivative of the considered output neuron with regard to the input features. Therefore,

[3] Note that $\mathbf{f}_l \in \mathbf{F}$ and $\mathbf{f'}_l \in \mathbf{F'}$. We consider these single input features separate from the combined input since we aim to create attributions for these inputs specifically. To obtain attributions for every input, we have to repeat the process for each input in the forecasting model.

saliency for input feature l with the combined input $\mathbf{F}$ into the neural network model, can be defined as

$$\text{Saliency}_l(\mathbf{F}) = \left| \frac{\partial \mathcal{O}_{\text{target}}(\mathbf{F})}{\partial \mathbf{f}_l} \right|, \tag{6.6}$$

where $\mathcal{O}_{\textbf{target}}$ is the target output neuron being explained, and $\mathbf{f}_l$ is the l-th input feature in the neural network whose effect on the model's output is being explained. As before, these inputs are either the last P history values or the next H steps of an exogenous feature forecast. Saliency relies on the assumption that the more influential a specific input is, the larger the gradient with regards to this input in the model and hence the larger the Saliency value [212].

Feature Permutation The first perturbation-based attribution method we consider is Feature Permutation (FP). Essentially, FP functions by shuffling the values for a single input feature, considering how this affects each of the model outputs [210]. More specifically, FP takes the considered l-th input feature vector and permutes the values within this vector. In our case, each feature vector contains multiple points in time, and we apply FP by randomly shuffling these points in time. This randomly shuffled feature vector is then used as an input to the model, and the difference between the considered output with permuted features and the original output is calculated. This difference is then considered as the feature attribution, i.e. the more the permutation affects the output, the more important the feature is. In our case, this permutation is performed both with regard to the mean forecast and the variance forecast. Therefore, we can define FP for feature l as

$$\text{FP}_l = \mathcal{O}_{\text{target}}(\tilde{\mathbf{F}}_l) - \mathcal{O}_{\text{target}}(\mathbf{F}), \tag{6.7}$$

where $\mathcal{O}_{\text{target}}$ is the output neuron considered, and $\tilde{\mathbf{F}}_l$ is the combined input of the neural network with a permuted feature $\mathbf{f}_l$. Whilst multiple permutations could be possible, we always consider random shuffling in the Captum implementation [234].

Feature Ablation The second perturbation-based attribution method considered is Feature Ablation (FA). In FA, attributions are calculated by replacing each input feature with a baseline and calculating the difference in the output. In the present chapter, we always consider the baseline as a vector of zeros. Therefore, we define FA for feature l as

$$\text{FA}_l = \mathcal{O}_{\text{target}}(\mathbf{F}'_l) - \mathcal{O}_{\text{target}}(\mathbf{F}), \tag{6.8}$$

where $\mathcal{O}_{\text{target}}$ is the output neuron considered, and $\mathbf{F}'_l$ is the combined input of the neural network where the l-th feature has been ablated and replaced with a baseline $\mathbf{f}'_l$, which is in our case always a vector of zeros.

Shapley Value Sampling The final perturbation-based attribution method considered is Shapley Value Sampling (SVS). SVS is based on the concept of Shapley values from cooperative game theory and essentially combines ideas from FP and FA. To apply SVS, a random permutation of the input features is taken, and each of these permutations is added, one by one, to a given baseline.

After adding each feature, the difference in the output is calculated, and this difference is used as the attribution from SVS [209]. This process is repeated numerous times with a different random permutation of features, and the final attribution is the average of the attributions for each feature across all permutations. A more detailed explanation of SVS is provided by Castro *et al.* [209] and Strumbelj and Kononenko [235].

6.3.3 Evaluation Metrics

To evaluate our approach, we consider forecast quality, which we measure with the Continuous Ranked Probability Score (CRPS) (see Equation (2.12)) and by considering prediction intervals. The main focus of our evaluation, however, is on the attributions created to explain both the mean and variance components of the probabilistic forecast. The majority of our evaluation focuses on visualising these attributions as heat maps to allow a comparison between the steps in the forecast horizon. However, to accurately compare attributions from different methods or at similar points in time, we require further evaluation metrics, which we present in the following.

Mean Scaled Absolute Attribution Difference To compare the attributions generated from different XAI methods we calculate the MSAAD. The MSAAD is based on the fact that we can only compare different attribution-based explainability methods if we first scale them to the same range. Furthermore, we want to compare the attributions across all samples in the test data set and not only for a single sample. Therefore, we define $A(n)$ as the attribution matrix obtained for a single sample n, for a given input feature. This matrix has the dimension $I \times H$, where $I = P$ for the history input and $I = H$ for the forecast exogenous inputs. Given this attribution matrix, we first calculate the scaled absolute attributions for the sample n as $\tilde{A}(n)$. More specifically, $\tilde{A}(n)$ is calculated by scaling the absolute value of the attributions per column so that the explanations for each output (i.e. forecast horizon) are between zero and one. With this scaling, the point in time with the highest influence obtains a value of one, and the point in time with the lowest influence has a value of zero. This can be defined as

$$\tilde{A}(n) = \frac{|A(n)_{\star,j}| - \min(|A(n)_{\star,j}|)}{\max(|A(n)_{\star,j}|) - \min(|A(n)_{\star,j}|)}, \text{ for } j \in 1, \ldots, H, \tag{6.9}$$

where $A(n)_{\star,j}$ refers to all rows in the j-th column, i.e. the j-th step in the forecast horizon, and $|\circ|$ implies the absolute value of each value in the attribution matrix. Given $\tilde{A}(n)$ for each sample n we can then compare attribution methods with

$$\mathrm{MSAAD} = \frac{1}{N} \sum_{n=1}^{N} \left[\frac{1}{I \cdot H} \cdot \mathrm{grandsum} \left(\left| \tilde{A}(n) - \tilde{A}'(n) \right| \right) \right], \tag{6.10}$$

where $\mathrm{grandsum}(\circ)$ is the matrix grand-sum operator, which involves adding all elements within the matrix to obtain a scalar, and $\tilde{A}'(n)$ is the scaled absolute attribution matrix generated by the alternative XAI method being compared. The MSAAD can be loosely interpreted as a mean average percentage error between attributions. Importantly, since we always scale the

attributions between zero and one, the MSAAD is only suitable to compare attribution methods and not to compare the attributions for different feature vectors. This scaling leads to all input features, including unimportant features, receiving the same weighting. For a pure comparison of the attribution methods such a scaling is sufficient, but this scaling does not enable a clear comparison of feature performance.

Temporally Similar Absolute Attributions To identify if similar attributions occur in time series with temporal patterns, we also consider temporally similar absolute attributions TSAA. Such attributes are calculated by considering similar points in time, i.e. all Mondays at midnight, and averaging the absolute attributions across all samples that correspond to this point in time. We, therefore, define temporally similar absolute attributions as

$$\mathrm{TSAA} = \frac{1}{|\Upsilon|} \sum_{\upsilon \in \Upsilon} |A(\upsilon)| \,, \tag{6.11}$$

where Υ is the set of temporally similar samples. The TSAA is in itself a $I \times H$ matrix that can also be visualised via a heat map. Note that we always consider the absolute value of the attributions to avoid averaging to zero if one sample demonstrates a strong negative attribution and the next a strong positive attribution.

Static Mean Absolute Attributions The previous evaluation metrics consider the temporal dynamics of the input features, i.e. they consider the attribution for each point in time in the input vector. However, it may also be interesting to investigate the static effect of a given input feature on the output, i.e. how important is the feature on average for a given output neuron. To investigate this aspect, we define the static mean absolute attributions SMAA, which is calculated by finding the average attribution value for each input feature. Specifically, for a sample n we can calculate the $\mathrm{SMAA}(n)$ as

$$\mathrm{SMAA}(n) = \frac{1}{I} \sum_{i=1}^{I} |A(n)_{i,\star}| \,, \tag{6.12}$$

where i is the dimension of the input vector being considered, with $I = P$ for the history input and $I = H$ for the forecast exogenous inputs. The $\mathrm{SMAA}(n)$ is then a $1 \times H$ vector and indicates the static influence of a given input for each of the output neurons for the considered forecast horizon. In the present chapter, we can calculate $\mathrm{SMAA}(n)$ for multiple samples in the test data set and visualise these samples as a $N \times H$ heat map. Furthermore, we must calculate $\mathrm{SMAA}(n)$ separately for the mean forecast and variance forecast. We apply this metric exclusively to better understand the attributions for the Solar data set.

Table 6.1.: Architecture of the applied probabilistic forecasting network. The main network processes the inputs before each output network generates forecasts for either the mean or variance of the probabilistic forecast.

(a) Main Network.

Layer	Description
Input	[Encoded Historical Information, Encoded Exogenous Forecasts]
1	Dense 512 neurons; activation: ReLU
2	Dense 256 neurons; activation: ReLU
3	Dense 128 neurons; activation: ReLU

(b) Output Network.

Layer	Description
Input	[Output of main network]
1	Dense 32 neurons; activation: ReLU
2	Dense H neurons; activation: linear

6.3.4 Probabilistic Forecasting Model Implementation

We realise the probabilistic forecasting network described in Section 6.2 with a simple feed-forward neural network. Thereby, each of the inputs is encoded via a single feed-forward layer. In this encoding process, the exogenous forecasts are encoded from the original input size H to a dense layer of 15 neurons, whilst the history input is encoded from the original input size P to a dense layer of 32 neurons. These encoded inputs are then concatenated and given as inputs into the main network. For the data sets where only historical information is considered, the number of exogenous forecasts is zero, i.e. $M = 0$, and therefore, this concatenation of inputs is not required. The encoded concatenated inputs are processed by a main network before two separate output networks generate the forecasts for the mean and variance. The architecture of this main network and the output networks are reported in Table 6.1.

To generate probabilistic forecasts, we assume a H-dimensional Gaussian distribution and train the network using the maximum likelihood principle [231]. Specifically, we use the negative loss likelihood function for the Gaussian distribution to train our network, i.e.

$$\mathcal{L}_{\text{NLL}}(\mathbf{y_{t+H}}, \hat{\boldsymbol{\mu}}_{\mathbf{t+H}}, \hat{\boldsymbol{\sigma}}^{\mathbf{2}}_{\mathbf{t+H}}) = (-1) \cdot \frac{1}{N \cdot H} \sum_{h=1}^{H} \sum_{i=1}^{N} \left[\frac{1}{2} \left(\frac{(y_{i,t+h} - \hat{\mu}_{i,t+h})^2}{2 \cdot \hat{\sigma}^2_{i,t+h}} \right) + \log \left(\sqrt{2 \cdot \pi \cdot \hat{\sigma}^2_{i,t+h}} \right) \right], \tag{6.13}$$

where $y_{i,t+h}$ is the true value from the i-th sample in the test data set for the forecast horizon $t+h$, with $h \in [1, 2, \ldots, H]$. Furthermore, $\hat{\mu}_{i,t+h}$ is the mean forecast for the i-th sample in the test data set for the forecast horizon $t+h$, and $\hat{\sigma}_{i,t+h}$ the corresponding variance forecast. The resulting negative loss likelihood is calculated for all samples across all forecast horizons. We train our models using the Adam optimiser [166], with a learning rate of 0.001 for 300 epochs.

The probabilistic forecasts are generated with a forecast horizon of 20 steps for the synthetic data ($H = 20$) and a forecast horizon of $24\,\text{h}$ for the real data sets ($H = 24$). Furthermore, for the

synthetic data, we always consider the last 40 steps of data as history input ($P = 40$). However, for the real data sets, we consider two different probabilistic forecasting models with different amounts of history input information. The first model uses the last $48\,\mathrm{h}$ of historical information as an input ($P = 48$), whilst the second variation uses the last $168\,\mathrm{h}$ ($P = 168$).

6.4 Evaluation

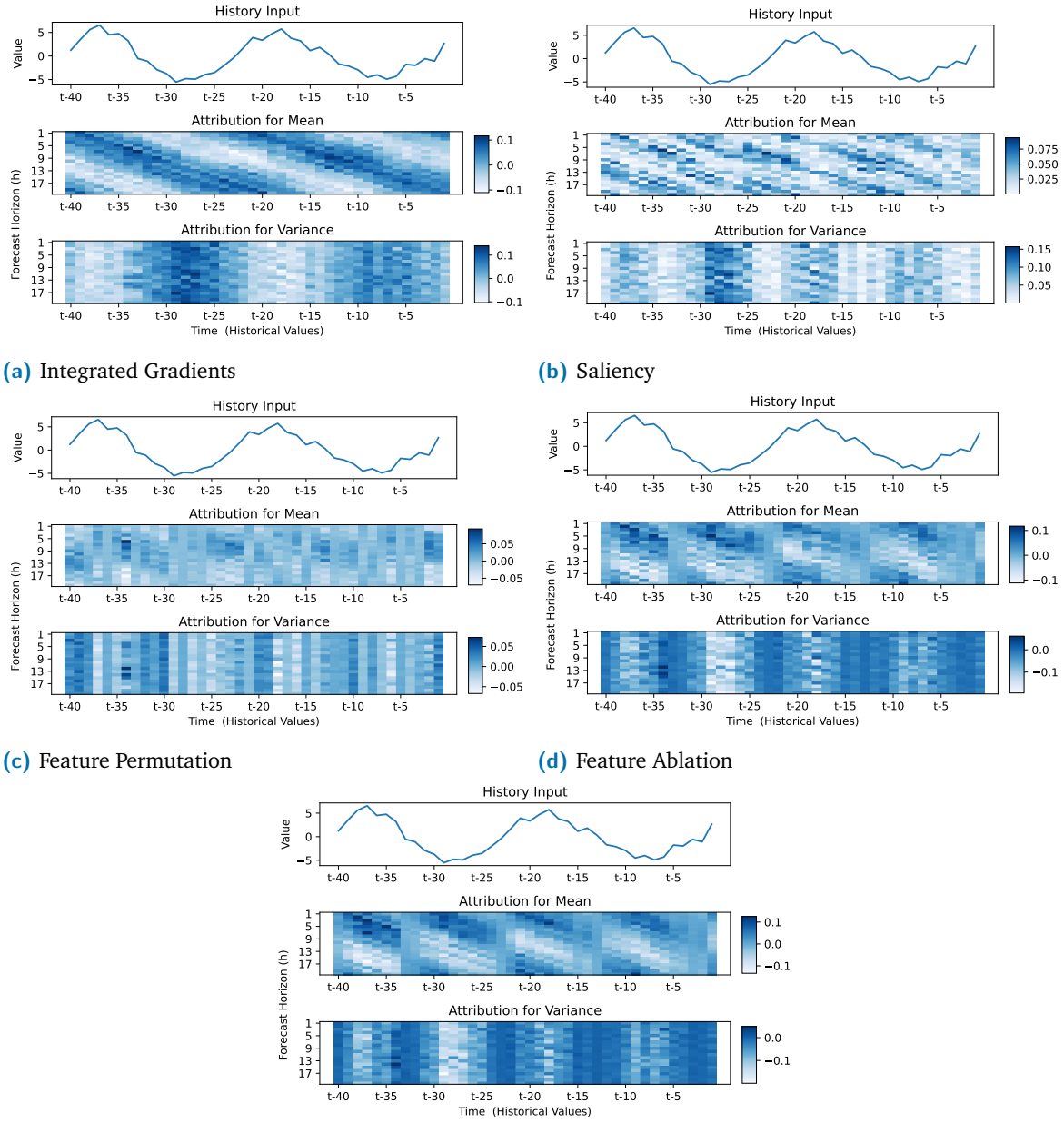
We consider three different aspects when evaluating our approach to explain the origins of uncertainty in probabilistic forecasts. First, we use the synthetic data set to visualise exemplary attributions from different XAI methods to analyse if any patterns can be identified. Second, we compare the attributions from different explanation methods. Finally, we analyse the attributions from the Saliency XAI method on the four real-world data sets to determine whether these provide useful information.

6.4.1 Exemplary Attributions

To compare attributions, we visualise the attributions for the history input obtained from different XAI methods for a single sample on the synthetic data with $\omega = 0.05$ in Figure 6.3. We first observe that although the attributions differ, there are also similarities. For the mean attributions, the history values that are a multiple of the period length result in the highest positive attributions, whilst those that are a multiple of half of the period result in the largest negative attributions. For example, for the output neuron for the forecast horizon of one, the history values approximately 20 steps and 40 steps back have high positive attributions, whilst those approximately 10 and 30 steps back have large negative attributions. However, for the output neuron for the forecast horizon of 10, the history values of approximately 10 and 30 steps in the past are strongly positive, whilst those 20 and 40 steps in the past are negative. These positive and negative attributions result in the diagonal pattern observed with varying degrees of clarity in all of the attribution methods. Importantly, Saliency considers the absolute value of the gradients and, therefore, only returns positive values. As a result, we cannot observe negative attributions in this case.

With regards to the attributions for variance, we also observe similarities across all the XAI methods. Namely, the peaks in the sine curve always result in the highest magnitude of attribution. This pattern is the clearest for IG and Saliency, although high magnitude negative attributions can also be observed in the peaks for FA and SVS. The attributions from FP are less clear, although the strongest attributions also seem to appear at the peaks.

Comparing the mean and the variance attributions, we observe a clear difference. Across the forecast horizons, different points in the history input are important for the mean forecast. However, for the variance forecast, the same points in the history input are important regardless of the considered forecast horizon. This observation is consistent for all considered XAI attribution methods.



(a) Integrated Gradients

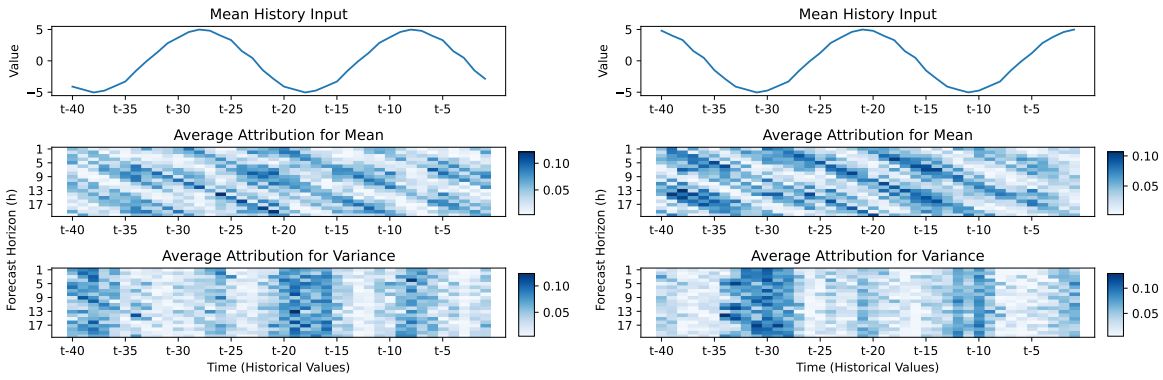
(b) Saliency

(c) Feature Permutation

(d) Feature Ablation

(e) Shapley Value Sampling

Figure 6.3.: A comparison of the attributions generated from the five considered XAI methods for a single sample on the synthetic data set with $\omega = 0.05$, i.e. a period of 20 steps. We consider the tenth sample in the test data set and apply different XAI to the same neural network. The network only considers the last 40 history values as input and generates a probabilistic forecast for the next 20 values. Although the attributions do differ, similar patterns can be seen for each method, with the clarity of these patterns varying.



(a) Offset of 0 **(b)** Offset of 13

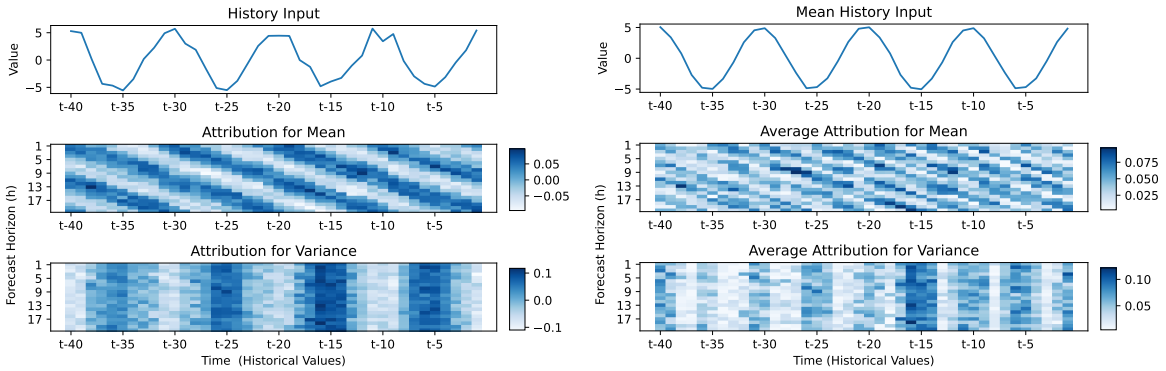
Figure 6.4.: A comparison of temporarily similar absolute attributes using the IG attribution method on the synthetic data with $\omega = 0.05$, i.e. a period of 20 steps. We compare two different temporally similar sets of attributions: the first considers all samples placed 20 steps apart, i.e. the length of one sine period, with an initial offset of zero from the beginning of the test data. The second set considers samples 20 steps apart with an initial offset of 13. The network only considers the last 40 history values as input and generates a probabilistic forecast for the next 20 values.

Since one sample is not representative of the entire time series, we visualise temporally similar absolute attributions for two different temporal groupings using IG on the synthetic data set with $\omega = 0.05$ in Figure 6.4. The temporal groupings are always separated by one-period length, i.e. 20 steps, and differ in the offset calculated from the start of the test data. Considering the temporally similar absolute attributions, we observe that the attributions for the mean forecast are almost identical. Even though the start offsets are different and the temporal groupings are at different points along the sine wave, we still observe similar patterns as for the single sample. Furthermore, the attributions for the variance forecast still focus on the peaks of the sine wave. However, since the offset causes the peaks to occur at different places, we see a clear difference in the attributions for the variance forecast between both temporal groupings.

To investigate whether similar patterns are observed on different data, we compare the attributions for a single sample and the temporally similar absolute attributes for an offset of zero using the IG XAI method on synthetic data with $\omega = 0.1$ in Figure 6.5. Specifically, doubling the frequency of the sine curve from $\omega = 0.05$ to $\omega = 0.1$ results in the period length being halved to 10 steps. This is reflected in Figure 6.5 where we observe an identical attribution pattern to before, except the frequency is doubled. Specifically, the diagonal pattern caused by the alternating positive and negative correlations occurs twice as often due to the sine wave oscillating twice as rapidly. Similarly, the attributions for the variance forecast still consider the peaks of the sine curve, with these peaks occurring twice as frequently.

6.4.2 Comparison of Attribution Methods

Although we observed similar patterns in the previous section, it was difficult to compare the XAI methods due to the different scales of the attributions. Specifically, some XAI methods



(a) Attribute for a Single Sample

(b) Temporally Similar Absolute Attributes, offset of 0

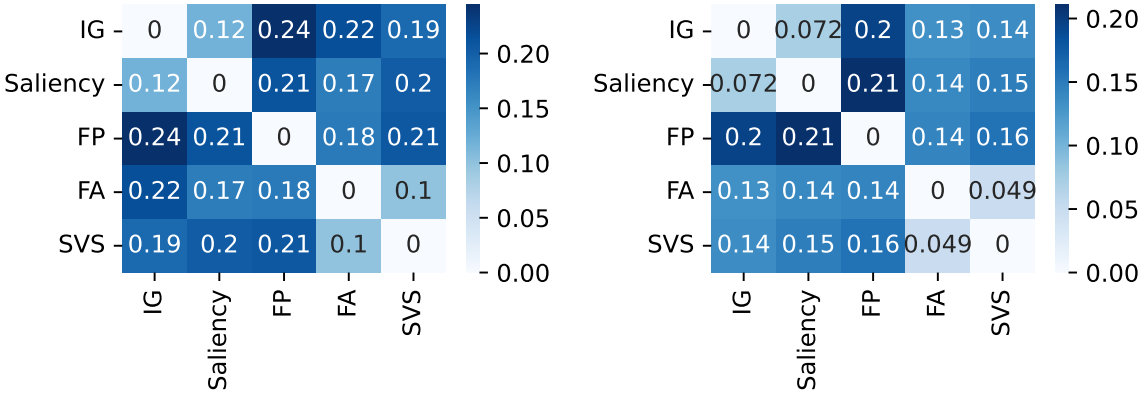
Figure 6.5.: A comparison of attributes using IG on the synthetic data set with $\omega = 0.1$, i.e. a period of 10 steps. We compare a single sample in (a), specifically the tenth sample in the test data set, and temporally similar absolute attributes for an offset of 0 in (b). The neural network only considers the last 40 history values as input and generates a probabilistic forecast for the next 20 values.

allowed for negative attributions, whilst others only considered positive attributions. Therefore, in this section, we consider the scaled absolute attributions to concretely compare the different attribution methods.

Specifically, to comprehensively compare the different XAI attribution methods, we calculate the mean scaled absolute attribution difference for all combinations and report these results on the synthetic data for $\omega = 0.05$ in Figure 6.6 and for $\omega = 0.1$ in Figure 6.7. We first observe that the results for both values of ω are almost identical. The second observation is that the attributions for the variance forecast are slightly more similar than those for the mean forecast. For example, with $\omega = 0.1$ the largest MSAAD for the variance is 0.21 between Saliency and FP, whilst the largest MSAAD for the mean forecast is 0.24 between IG and FP. Further, the smallest observed MSAAD is 0.05 for the variance forecast and 0.099 for the mean forecast. Third, we observe that there are two clear groups of methods that result in similar attributions. The first group is IG and Saliency with a MSAAD no larger than 0.12, measured for the mean forecast attributions on the synthetic data for both values of ω. The second group is FA and SVS whose MSAAD ranges from 0.05 to 0.1 depending on the quantity explained and the value of ω. Finally, we note that FP is the method with attributions that are the most different to all other methods. The lowest MSAAD involving FP is 0.14 when compared to FA on the synthetic data set with $\omega = 0.05$.

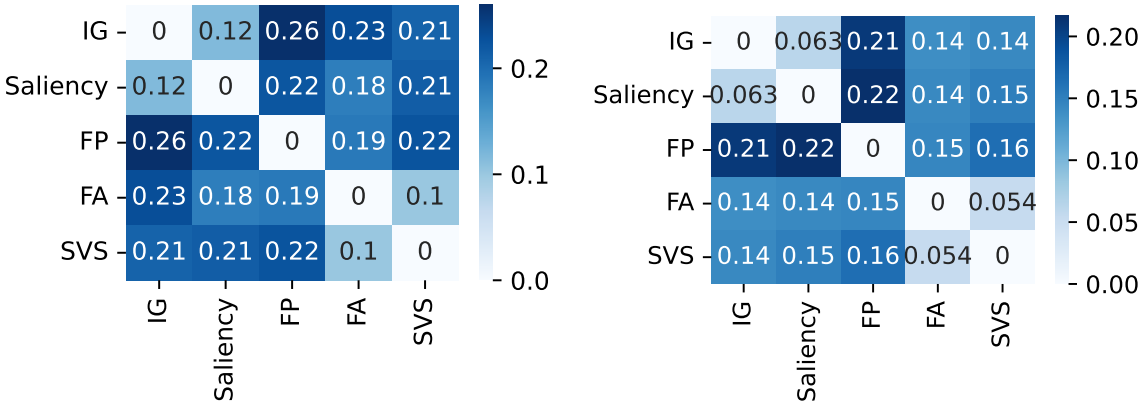
6.4.3 Analysis of Attributions on Real Data

In the final step of our evaluation, we apply our approach to real data to determine whether the explanations for the origins of the uncertainty provide us with insightful information. Thereby, we first briefly evaluate the forecast quality of our probabilistic forecasting models. Second, we analyse the attributions for the history input of all of our models. Finally, we consider the attributions for the exogenous forecast features for both the Price and Solar data sets. In this section, we only consider Saliency as an attribution method due to its similarity with IG and the



(a) Mean **(b)** Variance

Figure 6.6.: A comparison of the mean scaled absolute attribution differences for all attribution methods on the synthetic data with $\omega = 0.05$, i.e. a period of 20 steps. We have two groups of similar attributions: IG and Saliency form the first group, whilst FA and SVS form the second. The attributions from FP are the most different when compared to the other methods.



(a) Mean **(b)** Variance

Figure 6.7.: A comparison of the mean scaled absolute attribution differences for all attribution methods on the synthetic data with $\omega = 0.1$, i.e. a period of 10 steps. We observe almost identical results to those from the synthetic data with $\omega = 0.05$.

Table 6.2.: The average CRPS values calculated on the test data for all considered data sets. We compare two variations of our probabilistic forecasting model: the first model only considers the last $48\,\mathrm{h}$ of history input, whilst the second considers the last $168\,\mathrm{h}$. Both of these variants also consider exogenous features. All models generate probabilistic forecasts for a forecast horizon of $24\,\mathrm{h}$.

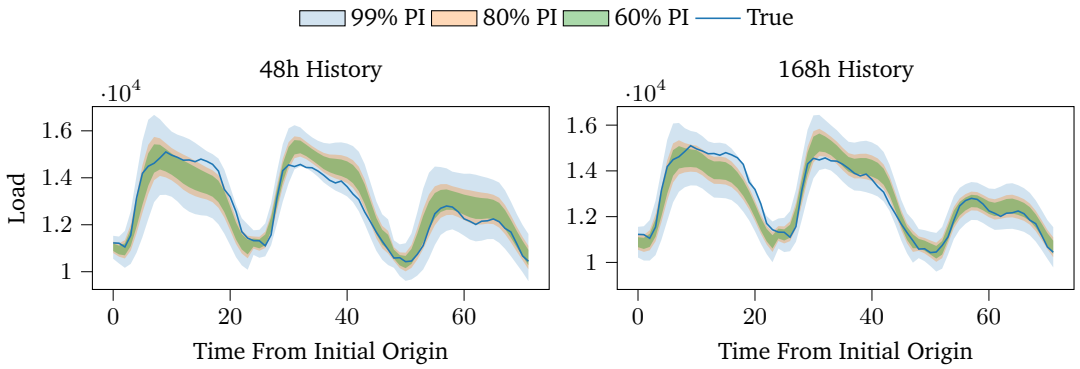
Model	Load Sweden	Load Germany	Solar	Price
History 48h	0.1080	0.1548	0.1236	0.1562
History 168h	0.0995	0.1137	0.1157	0.1777

reduced computational complexity compared to the perturbation-based approaches. Furthermore, we only present results for certain days and models and include further results in Appendix B.

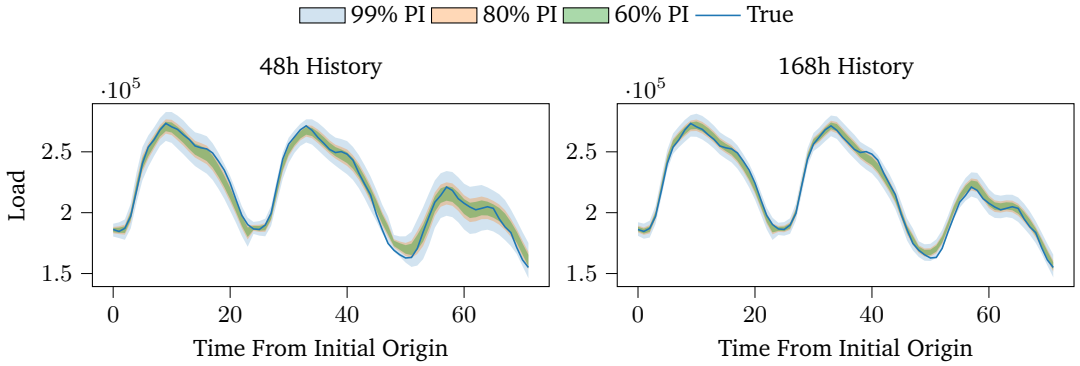
Forecast Quality Before analysing the attributions for the probabilistic forecasts, it is important to determine whether the models are generating reasonable probabilistic forecasts. Therefore, we briefly analyse the quality of the probabilistic forecasts by considering exemplary visualisations of the prediction intervals in Figure 6.8 and reporting the average CRPS for the test data in Table 6.2. Qualitatively considering the prediction intervals suggests that the generated probabilistic forecasts are of a reasonable quality. The prediction intervals are generally centred around the ground truth and appear both reasonably sharp and calibrated. Furthermore, we notice that the probabilistic forecasts from the models considering $168\,\mathrm{h}$ of history input appear better on all data sets except for Price. Noticeably, the model with only $48\,\mathrm{h}$ History appears to slightly underestimate all values on the Sweden load data set and overestimates the amount of uncertainty in the Germany load data set.

Considering the CRPS values in Table 6.2 confirms the observations from the prediction interval plots. Across all data sets, the worst performing model is the model with $168\,\mathrm{h}$ History on the Price data set with a CRPS of 0.1777. Although we cannot directly compare these results to Chapter 4 and Chapter 5 since we are using different input features and different train and test sets, this result is better than the majority of the benchmarks considered in those chapters. Similarly, low CRPS values for the remaining data sets confirm that the probabilistic forecasts are of a reasonably high quality. Finally, we confirm that the model with $168\,\mathrm{h}$ history input is indeed better on all data sets except Price, achieving a lower CRPS than the model with $48\,\mathrm{h}$ history input on the Sweden, Germany, and Solar data set.

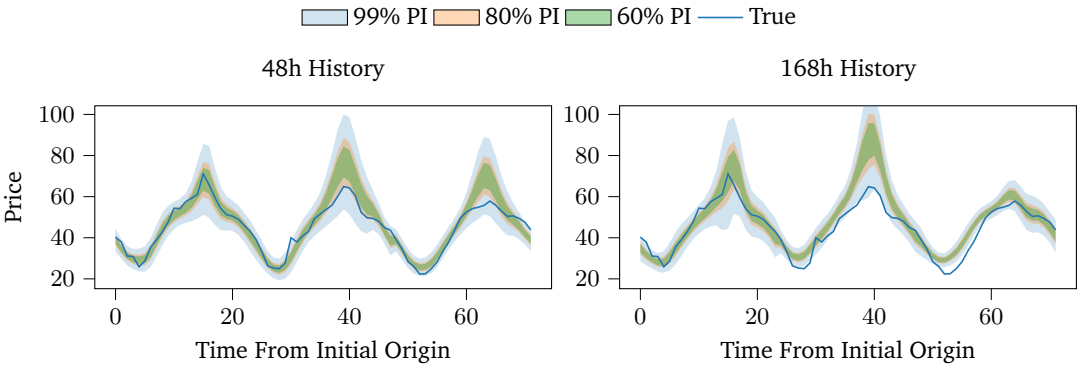
History Attributions To analyse the attributions for the history input feature, we consider each of the data sets individually. We begin with the Sweden load data set by visualising the temporally similar absolute attributions for Tuesdays in Figure 6.9. We first observe that in all cases, the most recent time step has the highest attribution. Second, for the mean attributions, we observe a slight diagonal pattern for the most recent $24\,\mathrm{h}$ of the history input for both models. However, this pattern is not as clear as the synthetic data. Furthermore, this pattern varies in strength depending on the day considered, i.e. it is stronger from Tuesday to Friday and weaker on Saturday, Sunday, and Monday (see Appendix B). Third, apart from the high attributions for the most recent time step and the slight diagonal pattern for the most recent day, the size of



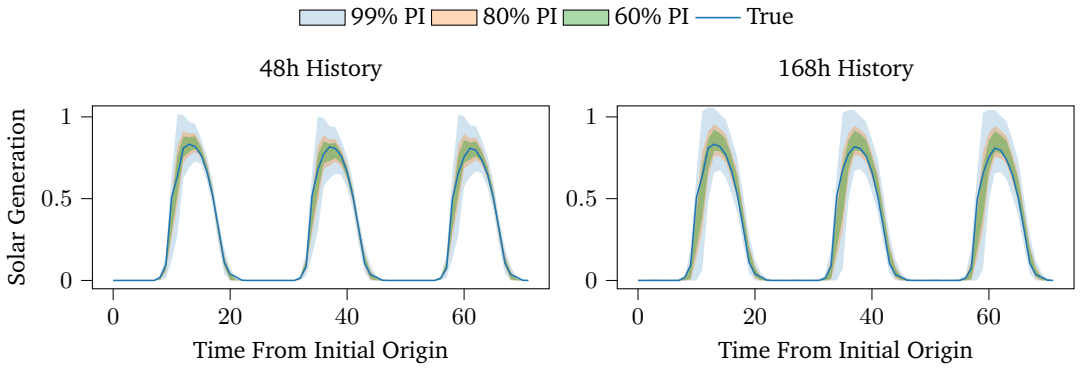
(a) Load Sweden, start at sample 316 in the test data



(b) Load Germany, start at sample 224 in the test data

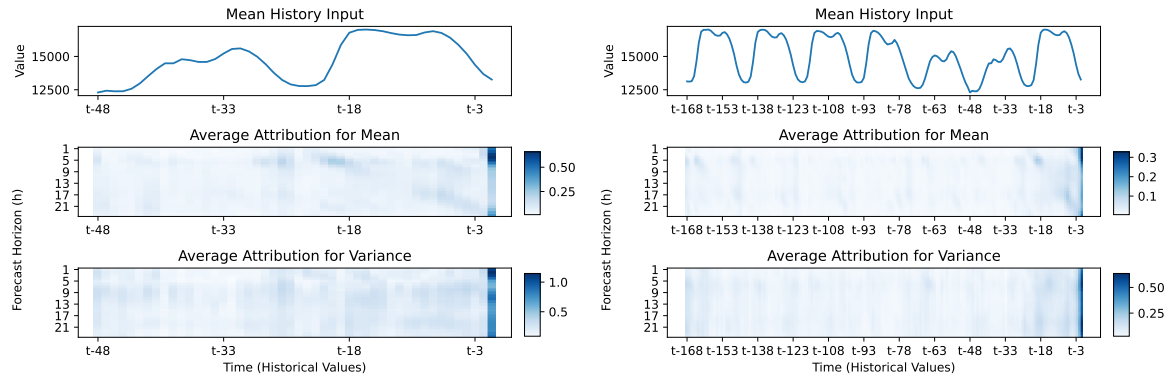


(c) Price, start at sample 1675 in the test data



(d) Solar, start at sample 169 in the test data

Figure 6.8.: Exemplary prediction intervals for both probabilistic forecasting models on all considered data sets. The forecast horizon is always 24 hours, i.e. $H = 24$, and in this figure, we plot three consecutive forecasts together starting from a random sample in the test data. We observe that for all data sets, the probabilistic forecasts appear reasonable, with the prediction intervals centred around the ground truth and being quite sharp and well-calibrated. The neural networks used consider either $48\,\mathrm{h}$ or $168\,\mathrm{h}$ of history values as inputs as well as exogenous features.



(a) 48 h History, Temporally Similar

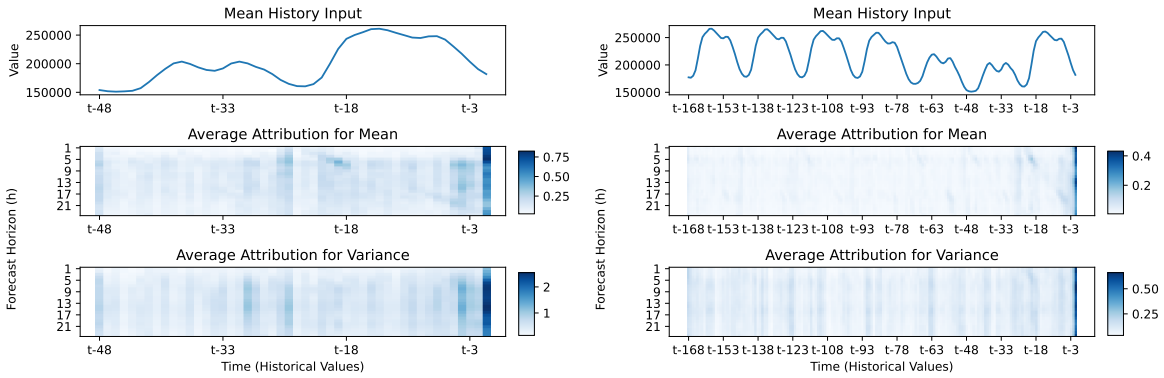
(b) 168 h History, Temporally Similar

Figure 6.9.: A comparison of the temporally similar attributions for Tuesdays on the Sweden load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

the attributions does not vary noticeably for history inputs further in the past. Instead, these attributions fluctuate over the entire history input. Fourth, we observe a high correlation between the attributions for the mean forecast and the variance forecast. Specifically, when considering the temporally similar attributions, the attributions for the mean and the variance forecast are almost always identical.

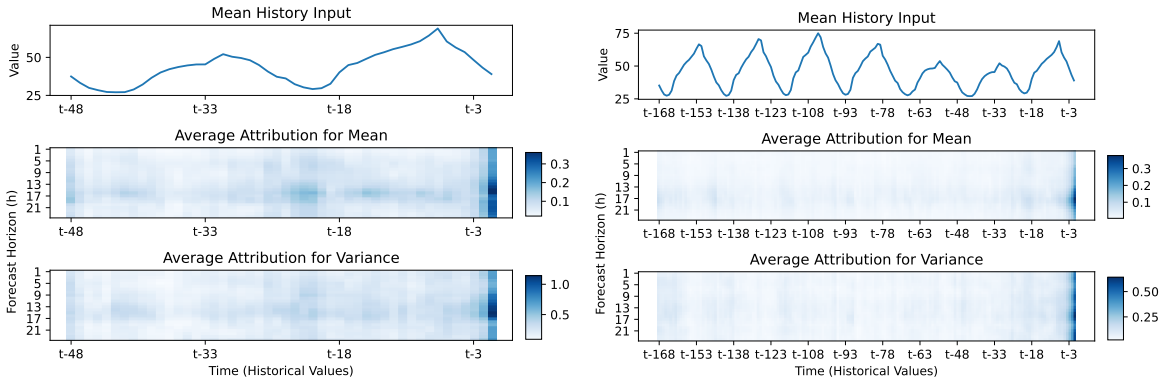
We also visualise the temporally similar absolute attributions for Tuesdays for the Germany load data in Figure 6.10. On the Germany load data set, we again observe that the most recent history input receives the largest attribution for all models for both the mean and variance forecast. Second, we again observe a slight diagonal pattern in the mean attributions for the most recent 24 h for both models. If anything, this diagonal pattern is slightly more pronounced than the Sweden load data. Third, we again observe that the attributions for the mean and variance forecast are similar, however, they differ more than the Sweden load data. For example, for the model considering 168 h of historical information, a reoccurring pattern approximately ever 12 h can be seen for the variance attributions. However, despite these minor differences, the temporally similar absolute attributions for the Germany load data set are similar to those for the Sweden load data set.

The attributions vary slightly when we consider the temporally similar absolute attributions for Tuesdays for both models on the Price data in Figure 6.11. We first observe that the most recent value is still obtaining the highest attribution for both the mean and variance forecast for both models. Also, as with the previous data sets, the attributions for the mean and variance forecast are almost identical. However, unlike the previous data sets, we do not observe any diagonal pattern over the last 24 h. Instead, it appears that both models place slightly more importance on the value exactly 24 h in the past since this point in time results in a slightly higher attribution for both models. Furthermore, this higher attribution is visible in both the mean and variance attribution. However, it is worth noting that this increased size of the attributions is only marginal.



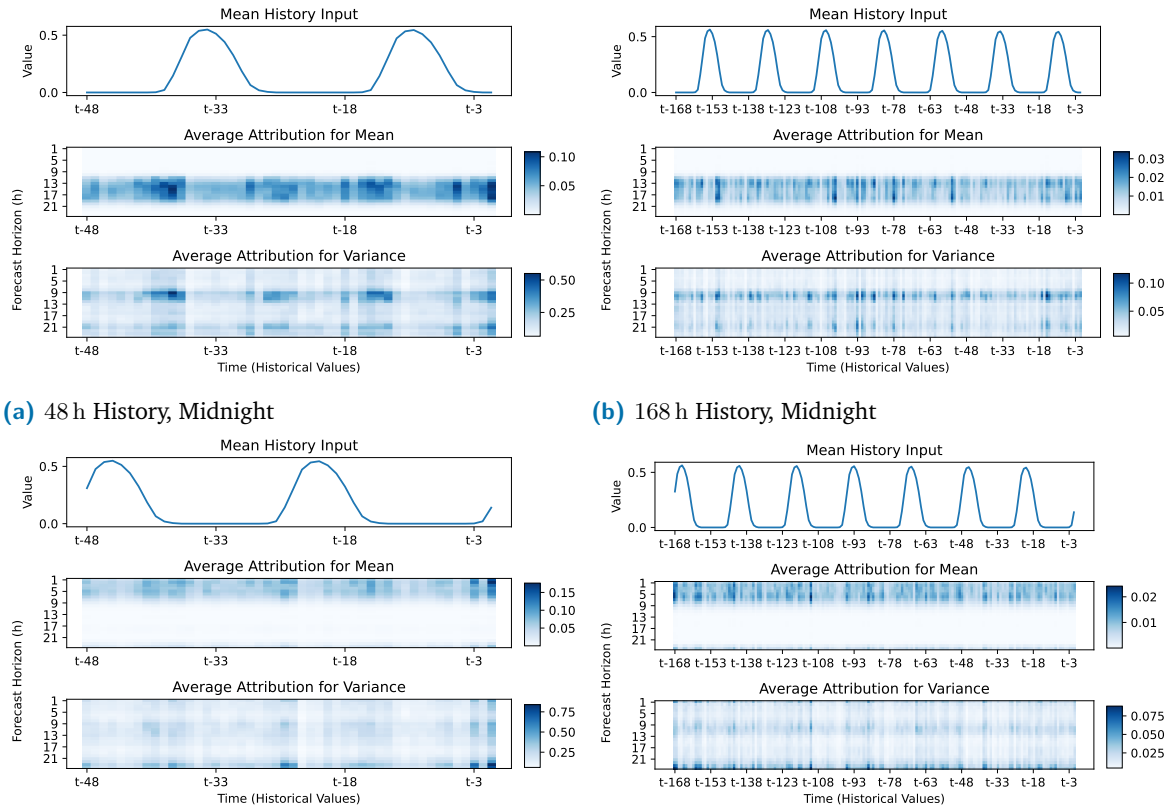
(a) 48 h History, Temporally Similar **(b)** 168 h History, Temporally Similar

Figure 6.10.: A comparison of the temporally similar attributions for Tuesdays on the Germany load data. Both the models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.



(a) 48 h History, Temporally Similar **(b)** 168 h History, Temporally Similar

Figure 6.11.: A comparison of temporally similar absolute attributions for the history input on Tuesdays between the model with 48 h History and the model with 168 h History on the Price data. The neural network also considers exogenous features as an input and always generates a probabilistic forecast for the next 24 h.



(a) 48 h History, Midnight

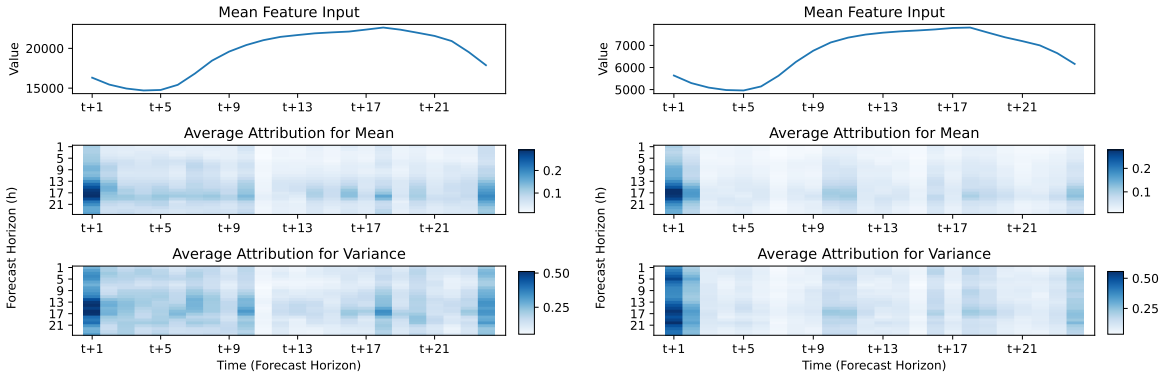
(b) 168 h History, Midnight

(c) 48 h History, 11 am

(d) 168 h History, 11 am

Figure 6.12.: A comparison of temporally similar absolute attributions at two different times of the day on the Solar data set for the two forecasting models considering 24 h and 168 h of history inputs respectively. The neural network also considers exogenous features as an input and always generates a probabilistic forecast for the next 24 h. We observe that the model only considers the history information for points in time when solar generation is expected, i.e. when it is daytime.

In contrast to the previous data sets, the history attributions for the Solar data set are noticeably different. Furthermore, unlike the previous data sets, solar data is characterised by daily patterns and not weekly patterns, therefore, we visualise the temporally similar absolute attributions for two different times in a day in Figure 6.12. The first observation is that for many forecast horizons, the history input for the mean forecast is completely ignored, i.e. we obtain an attribution of zero. For the temporally similar attributions at midnight, the forecast horizons for the mean forecast that consider the history input are between 9 h and 19 h ahead. On the other hand, the temporally similar attributions at 11 am only consider the history input for the mean forecast for the first 8 h steps ahead and then again from 22 h ahead. This suggests that the history values are only relevant for the mean forecast for forecast horizons that occur during daylight hours. Furthermore, within these attributions, we don't identify any clear patterns. If anything, there is perhaps a slightly higher attribution for the mean forecast at the beginning of a history day, i.e. as solar power begins to be generated each day and before the spike of peak generation occurs. However, on a whole, the attributions for the mean forecast are consistently high across the entire history input for those forecast horizons that are considered. Importantly, unlike the previous data sets, the most recent value does not receive the highest attribution.



(a) Total Load **(b)** Zonal Load

Figure 6.13.: A comparison of the temporally similar absolute attributes for Tuesdays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers $48\,\mathrm{h}$ of historical information as well as the exogenous features for the forecast horizon and generates a forecast for the next $24\,\mathrm{h}$.

With regard to the attributions for the variance forecast, we observe that these are noticeably different from the attributions for the mean forecast. For both models and both considered times of day, the history input feature is considered for every forecast horizon. However, despite the attributions being visible for more forecast horizons than for the mean forecast, the visible patterns are similar. More specifically, where the attributions fluctuate for the mean forecasts, they also fluctuate for the variance forecast, but for the variance, these fluctuations are visible for all forecast horizons. Therefore, it is also difficult to identify a clear correlation between the fluctuations and the structure of the mean history input for the variance forecast attributions. As with the attributions for the mean forecast, the most recent values also do not always obtain the highest attributes.

Exogenous Attributions The previous results have only considered the attributions for the history input, however, two of the considered data sets also contain forecast exogenous features as inputs. We visualise the temporally similar attributions for Tuesdays for the two exogenous features for the Price data set with the model considering $48\,\mathrm{h}$ of historical information as an input in Figure 6.13. Similar to the history input, the highest attribution for both features is always the first value in the forecast horizon, i.e. the value closest to the forecast origin. Second, for both features, we observe that the attributions for the mean forecast and the variance forecast are almost identical. Third, the attributions for both features are similar, although this similarity is not surprising since the considered features demonstrate similar patterns and differ only in the scale. Fourth, we do not observe any diagonal pattern in the input, which would be expected if the model was accurately mapping the forecast horizon of the input to the forecast horizon of the output target.

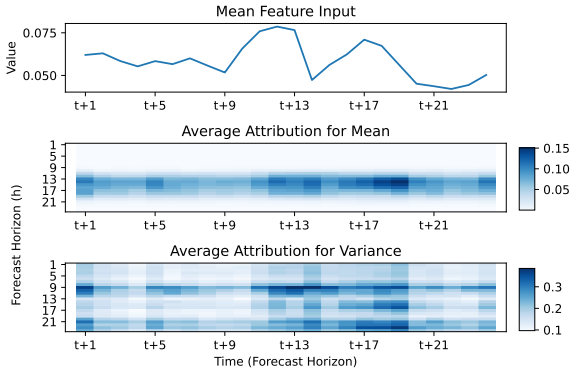
For the Solar data set, we visualise the temporally similar attributes for the three exogenous features at midnight using the model taking $48\,\mathrm{h}$ of history input in Figure 6.14. For all three exogenous features, we again observe that only certain forecast horizons are considered important by the model for the mean forecast, with other forecast horizons receiving attributions of zero.

As with the history inputs, the important forecast horizons for the mean forecast are between $9\,\mathrm{h}$ and $19\,\mathrm{h}$ ahead when the forecast origin is midnight, and for the first $8\,\mathrm{h}$ steps ahead and then again from $22\,\mathrm{h}$ when the forecast origin is 11 am. Second, we again observe that the variance attributions occur for all forecast horizons, however, the patterns observed in the variance forecasts are again similar to those in the mean forecasts. Third, although there are fluctuations across the forecast horizon for all features, these fluctuations do not seem to react clearly to the input feature's patterns. For example, for SSRD and the model considering $48\,\mathrm{h}$ of historical information (e), there does appear to be a higher attribution for the mean forecast when SSRD is high between 9 am and 6 pm, however, there is also a high attribution between 1 am and 3 am. Finally, if we compare the values of the attributions for the different features, we note that these values vary across the input features. More specifically, if we consider the attributions for the mean forecast, the TCLW feature obtains a maximal attribution of 0.15, the TCC feature a maximal attribution of 0.075 and SSRD a maximal attribution of 0.2. These differences are more noticeable for the attributions for the variance forecast, ranging from a maximum of 1.0 for the SSRD feature down to a minimum of 0.2 for the TCC feature.

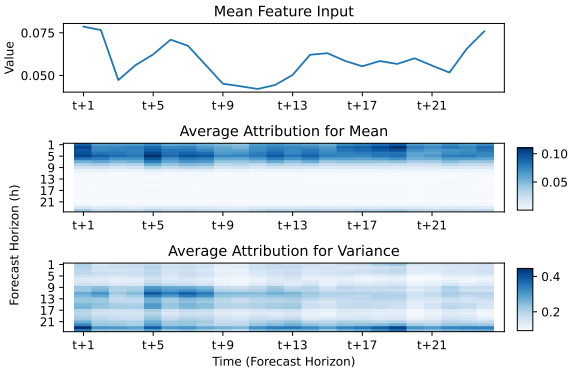
To gain further insight into the attributions for the Solar data, we consider the Static Mean Average Attributions for the first three days of the test data in Figure 6.15. The SMAA considers the mean absolute effect of an input feature on the considered output, i.e. it shows the average absolute attribution of this feature for the $24\,\mathrm{h}$ of considered input. For example, SMAA History indicates how important the history input is, on average, for each forecast horizon given a specific sample. In Figure 6.15, each sample is a new forecast origin, whereby we move forward one hour for each forecast. This visualisation clearly shows that the inputs for both models are only relevant for certain forecast horizons. Furthermore, the diagonal pattern highlights how these relevant forecast horizons change depending on the forecast origin and are almost always daylight hours. Consider, for example, the SMAA for SSRD. At 1 am, the SSRD inputs are relevant for a forecast horizon between $8\,\mathrm{h}$ and $21\,\mathrm{h}$. However, at 4 pm the SSRD inputs are only relevant for forecast horizons from $1\,\mathrm{h}$ to $6\,\mathrm{h}$ and then again from $17\,\mathrm{h}$ to $24\,\mathrm{h}$. Whilst we also observe a diagonal pattern for the variance attributions, this is not as clear as the attributions for the mean forecasts. Another interesting insight from the SMAA is that we can use them to see which features, as a whole, are important for the model. Since the attributions are scaled across all inputs into the model, the darker the attributions, the higher that features mean impact for the model. Here, we clearly observe that for both the mean and variance forecasts, the darkest attributions are for the SSRD feature. This implies that this feature, on average, receives the highest attributions and is, therefore, more important for the model.

6.5 Discussion

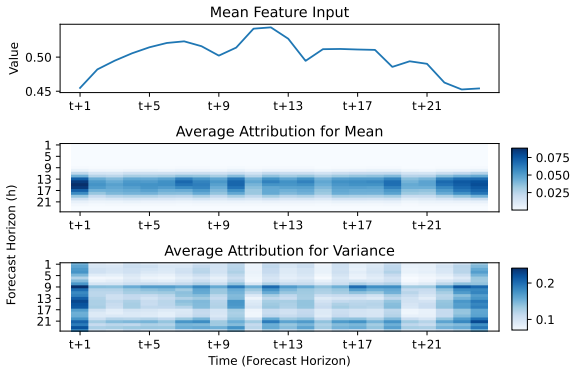
In this section, we discuss the evaluation of our approach to explain the origins of uncertainty in probabilistic forecasts. We first discuss some key observations from the results before highlighting the limitations and benefits of our approach.



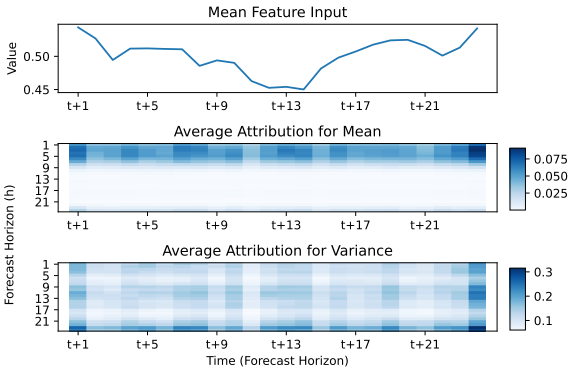
(a) TCLW, Midnight



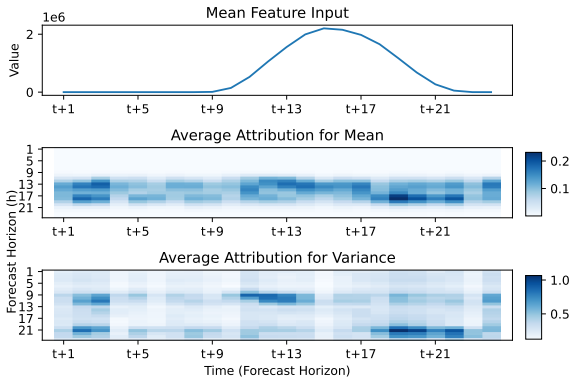
(b) TCLW, 11 am



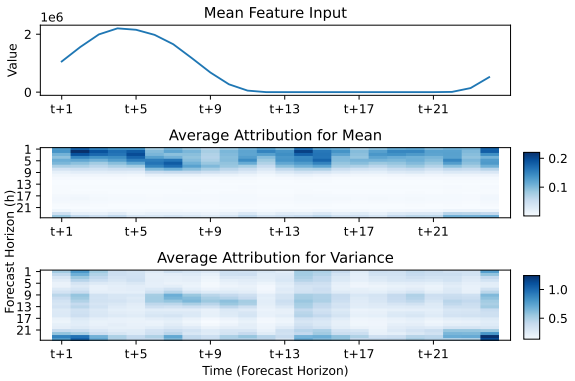
(c) TCC, Midnight



(d) TCC, 11 am

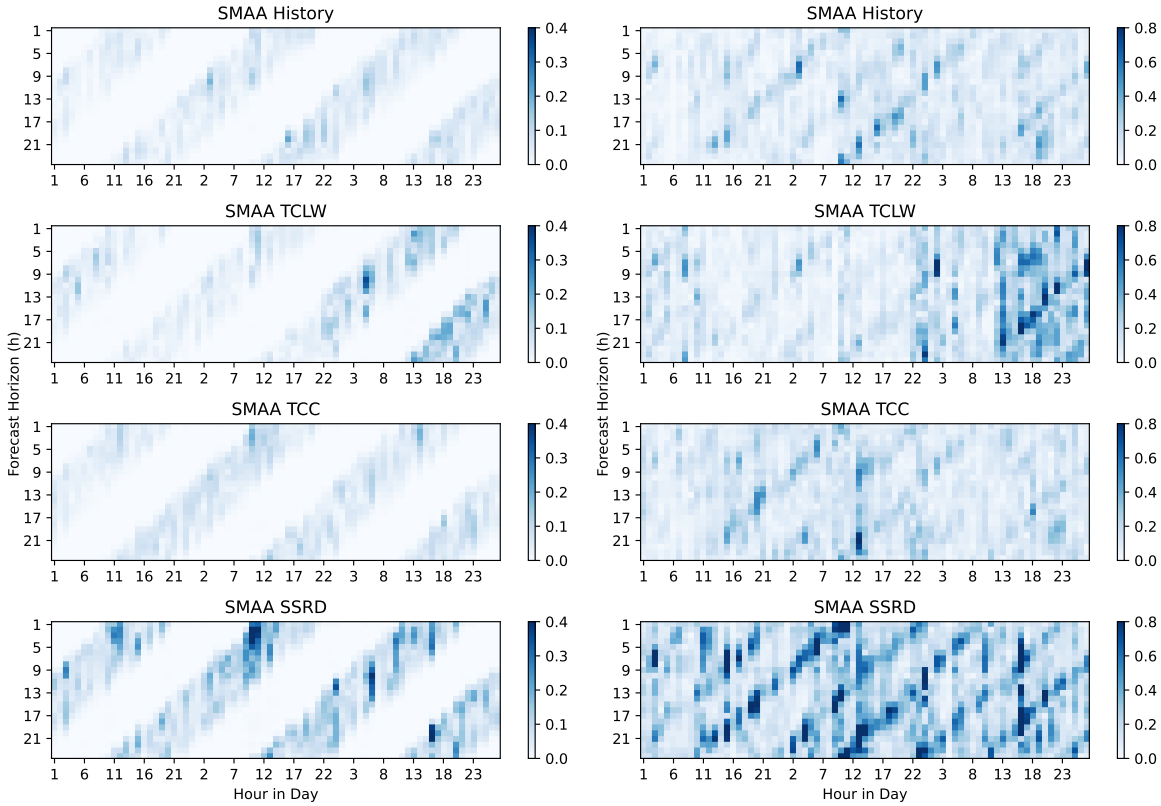


(e) SSRD, Midnight



(f) SSRD, 11 am

Figure 6.14.: A comparison of the temporally similar absolute attributes at midnight and 11 am for the three exogenous forecast features for the Solar data set. The comparison is created using the model that considers $48\,\mathrm{h}$ of historical information as well as the exogenous features for the forecast horizon and generates a forecast for the next $24\,\mathrm{h}$.



(a) Attributions for the mean forecast

(b) Attributions for the variance forecast

Figure 6.15.: The Static Mean Average Attributions (SMAA) for each input feature for the Solar data set using the model considering 48 h of historical information. The diagonal pattern for each feature indicates that only forecast horizons that occur during the day consider the features. Furthermore, the darker colours for SSRD indicate that this feature has the highest attributions on average and is, therefore, the most important feature for the forecast.

Results The first main observation is that on the synthetic data, our proposed approach is capable of generating attributions for both the mean and variance forecast that seem plausible. Importantly, these attributions are noticeably different, suggesting that the models consider different points in time in the history input when estimating uncertainty. More specifically, for the mean forecast attributions, our approach suggests that the model places more relevance on values that are multiple periods of the sine wave in the past. However, the attributions for the variance forecast demonstrate that the model considers the peaks of the sine wave when estimating the uncertainty. One possibility why this is plausible is that the peaks of the sine wave demonstrate the extreme values which are most likely to be affected by noise, i.e. if an unexpected spike occurs at a peak, then this will have a more noticeable effect on the sine curve than a spike occurring in the middle of the wave. However, although these initial results appear plausible, it is important to investigate more complicated synthetic data sets with more complicated uncertainty structures in the future.

Second, we observe that although the explanations from multiple attribution-based XAI methods differ, they still demonstrate similar patterns in the attributions on the synthetic data. A direct comparison of the attribution-based method highlights that these similarities are higher for two groups of methods. The first of these groups is Saliency and IG, which are also the two gradient-based attribution methods. The second group is FA and SVS, which are both perturbation-based methods. However, although FP is also a perturbation-based method, this method is not grouped with the others as it generates the most different attributions. It may, therefore, be interesting to further investigate FP and the reasons for its differing attributions in future work.

Third, for the two real-world data sets that only consider history input, we observe slight diagonal patterns for the mean attributions, but these patterns cannot be observed for the history input for the models that also consider exogenous inputs. Furthermore, when we consider the exogenous inputs for these models, we cannot observe any diagonal pattern in these inputs either. Such diagonal patterns suggest that the model is mapping the appropriate history value or exogenous feature value to the forecast output neuron, and their absence suggests that the models are not effectively learning this mapping. One explanation is that the combination of history inputs and exogenous feature inputs in a relatively simple network structure is too complicated for the network to effectively learn. Therefore, it is important to analyse whether more advanced model structures or similar model structures with a reduced number of features lead to clearer patterns in the attributions.

Fourth, we observe that, in general, the attributions obtained for the real-world data are similar for both the mean forecast and the variance forecast. Furthermore, these attributions suggest that the most recent value for the history inputs is by far the most relevant for three of the four data sets. These results seem to further suggest that the applied models do not accurately consider the temporal structure of the input when generating forecasts and only focus on the most recent value and a few singular points further in the past. To further investigate this phenomenon networks that are better suited to capture temporal dynamics, e.g. long short-term memory networks, should be applied and their outputs explained.

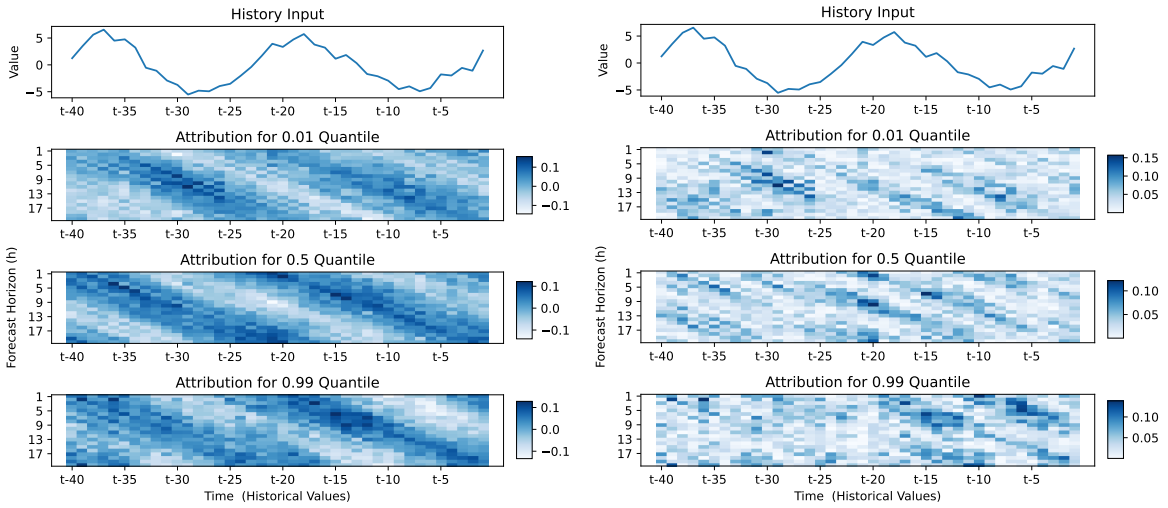
Fifth, the attributions obtained for the mean forecast on the Solar data set suggest that the model only considers the inputs for forecast horizons occurring during daylight hours. Furthermore, the model does not clearly react to spikes in the input from either the history or exogenous features. This suggests that either the model is not accurately learning the temporal dynamics in the input, or the input data is not presented in a way that allows these temporal dynamics to be learnt. Therefore, in addition to alternative models, it would be interesting to investigate if smart data representations can improve the model's ability to learn temporal dynamics, similar to Neumann *et al.* [236].

Sixth, the Static Mean Average Attributions present for the Solar data provide valuable information regarding overall feature performance. We observe that various inputs do receive different-sized attributions when compared to one another. For the Solar data set, the solar surface radiation downwards is consistently identified as the most important feature. Since solar radiation is the main factor contributing to solar power generation, this attribution is also logical. Therefore, such SMAA may be a valuable tool when selecting features in model design.

Limitations Despite the interesting results, our approach has some limitations. Firstly, we only compare common attribution-based explanations, mostly from the computer vision domain. However, the temporal dynamics of time series pose different challenges, and therefore, it would be interesting to develop attribution-based methods that are specifically designed for time series, e.g. extending approaches for time series classification developed by Tonekaboni *et al.* [221] and Munir *et al.* [222].

Secondly, the heat map visualisations presented are not always directly informative or intuitive. For the synthetic data, for example, plausible explanations exist as to why the model considers the peaks in more detail for the variance attributions, however, it is also possible to construct arguments as to why this is not logical. For example, since the peaks indicate the extreme values within the synthetic data, it is plausible that these points are responsible for the uncertainty. However, on the other hand, since the noise applied to the synthetic data is equal for all points in time, it is possible to argue that the explanation for the uncertainty should be equally weighted for all time steps. Furthermore, for the real-world data, it is difficult to judge whether the high attributions for specific points in time are logical without domain knowledge. The lack of intuitive information when using heat maps to visualise time series attributions for classification tasks was previously highlighted by Rojat *et al.* [49], however, currently no alternative has been proposed. Therefore, it would be worthwhile considering a more informative way of visualising and communicating attributions, perhaps also using smart data representations [236].

Third, our approach relies on separate forecasts for the deterministic and uncertain components of the forecast. Non-parametric methods, such as quantile forecasts, do not automatically contain this separation, and therefore, our approach can not yet be applied to explain the uncertainty in such models. This is demonstrated in Figure 6.16, where we compare the attributions for three different quantiles on the synthetic data set. We see that the attributions are almost identical



(a) IG **(b)** Saliency

Figure 6.16.: A comparison of the attributions generated Saliency and IG for quantile forecasts for a single sample on the synthetic data set with $\omega = 0.05$, i.e. a period of 20 steps. We consider the tenth sample in the test data set and use a neural network that only considers the last 40 history values as input and generates a probabilistic forecast for the next 20 values. Although the quantiles are fundamentally different, the attributions for each quantile are almost identical.

despite fundamentally different quantiles being forecast. Therefore, future work should also investigate how the origins of uncertainty can be explained in such non-parametric models.

Fourth, all our evaluations in the present chapter focus on time series that are predictable, i.e. a simple sine curve, or real-world data with clear patterns or helpful exogenous features. Therefore, we expect that each input feature is important for the output and thus receives an attribution. However, this may not be the case if we were to consider a data set comprised of random data with no useful features. Therefore, it would be interesting to investigate the attributions obtained when a forecasting model is trained on such completely random data.

Finally, in the present chapter, we explain the origins of uncertainty in probabilistic forecasts via post-hoc attribution methods, however, another possibility is to create probabilistic forecasts that are directly interpretable. Although recent work does attempt to consider interpretability when designing forecasting models, these approaches are almost entirely focused on point forecasts [40], [213], [214], [223], [224]. Therefore, further investigating probabilistic forecasting models that are directly interpretable, specifically regarding the origins of uncertainty would be worthwhile. Ideally, such interpretable forecasting models will generate probabilistic forecasts with a similar quality to state-of-the-art probabilistic forecasts, thus removing the need for post-hoc explanations [237]. Therefore, these interpretable models should be compared to state-of-the-art probabilistic forecasting models both regarding forecast quality and the effect of their interpretable design on trust.

Benefits Our approach demonstrates several key benefits. The first benefit is the ability to create explanations for both the deterministic and uncertain components of a probabilistic forecast with

existing XAI methods. These methods can be applied to a variety of neural networks, with the only requirement being that the gradients of the model can be propagated back to the inputs. In fact, for the perturbation-based attribution methods, even this requirement must not be fulfilled. As a result, our approach can easily be applied to existing probabilistic forecasting models without modifying these models to make them interpretable.

Second, the explanations provide valuable information for model development. Comparing the attributions for all input features, as in Figure 6.15, can be used to determine feature importance and possibly perform feature selection in the model design process. This is particularly useful if, as with the synthetic data, different regions of the input are important for the mean and variance forecasts. In such a case, one feature may be irrelevant for the mean forecast but important for the variance. Therefore, future work could explore how the attributions from our approach can be extended to improve model design.

Finally, our approach highlights possible challenges in the model architecture. For example, our explanations suggest that the considered models struggle to learn the temporal dynamics of the real data, and specifically for the Solar data, the dynamics in the input are completely ignored. Such observations are vital for deploying forecasting models in real-world settings. In this case, the explanations suggest the models are poorly designed, contrary to the low CRPS scores suggesting high-quality forecasts. As a result, the end user will be encouraged to not blindly trust the forecasts generated from the model. Therefore, by combining attributions with forecast quality evaluation, forecasting models can be designed that react appropriately to inputs and demonstrate high forecast quality, i.e. truly trustworthy forecasting models.

6.6 Conclusion

To increase trust in probabilistic forecasts, we present an approach that explains the origins of uncertainty in these forecasts. We first separate the deterministic and uncertainty components of the forecast through a model architecture with two separate outputs forecasting both the mean and the variance of the probabilistic forecast. Given these separated outputs, we explain existing attribution-based XAI methods to generate attributions for each point in time for each of the model inputs. By visualising these inputs as heat maps, it is possible to determine which regions of the inputs are responsible for the uncertainty in the resulting probabilistic forecast.

We evaluate our approach by comparing multiple attribution-based XAI methods on synthetic data and show that our approach is capable of explaining both the mean and variance forecast. Furthermore, these explanations appear plausible, highlighting periodic values for the mean forecast and extreme regions for the variance forecast. Furthermore, we apply our approach to four real data sets, generating explanations for both the history input and exogenous forecast features. This application reveals key information about the models, such as the relative importance of exogenous features and their failure to learn temporal dynamics in the input data. As a result, these explanations can be used as a further tool to evaluate the trustworthiness of a model before

deploying it in real-world situations and can also provide valuable insights for further model development.

In light of these interesting results, there are numerous opportunities for future work. Firstly, our approach should be extended to state-of-the-art probabilistic forecasting models, which should be better suited to capturing temporal dynamics. Second, explainability methods specifically designed for time series applications should be extended and integrated into our approach. Third, alternative visualisation methods that are intuitive for time series should be developed to better communicate the information from our explanations. Fourth, applying our approach to optimise feature selection and develop intuitive models should be considered. Finally, our approach should be extended to explain the uncertainty in non-parametric forecasts where the uncertainty component cannot be simply separated.

7 Representing Critical Regions of Uncertainty for Mobility Applications

The content of this chapter is based on:

K. Phipps *et al.*, "Customized uncertainty quantification of parking duration predictions for EV smart charging", *IEEE Internet of Things Journal*, pp. 1–1, 2023. DOI: 10.1109/JIOT.2023.3299201.

As human mobility is faced with growing changes due to, for example, climate change [238] and an increase in remote work [239], innovative solutions are required to adapt to these changing patterns and develop a mobility system that is efficient and sustainable for many years to come [240]. However, to realise such a system, it is important to quantify the uncertainty in many different applications. As a result, probabilistic forecasts for many important mobility applications already exist in the literature, for example, for ride-sharing and taxi demand [241]–[244], for traffic congestion [245], [246], for bike-sharing services demand [247], [248], for public transport demand and public transport headway [249], [250], and for parking space availability [251], [252].

However, merely generating these probabilistic forecasts will not create a sustainable mobility system. Instead, the probabilistic forecasts should be used by humans or autonomous systems, such as smart traffic signals [240], [253], to make decisions that will enable these sustainable mobility systems. Depending on the mobility application, the importance of certain regions of uncertainty will change. In bike-sharing, for example, it is crucial that peak demand is met, otherwise, the bike-sharing service will lose customers. On the other hand, for public transport operators, the forecast headway used for timetable creation should never be overestimated since this may result in public transport departing earlier than expected and passengers missing their commute. As a result, the quantified uncertainty should be represented in a way that accounts for important regions and assists decision-makers. For example, the representation of the uncertainty quantification in bike-sharing demand should specifically account for the uncertainty surrounding peak demand to avoid underestimation. Furthermore, the representation of uncertainty for headway forecasts should specifically help decision-makers avoid overestimation.

Therefore, in the present chapter, we consider customised representations of uncertainty to specifically assist mobility applications. Specifically, we focus on the application of Electric Vehicle (EV) smart charging and the uncertainty associated with the time an EV user spends in a given

location, i.e. the *parking duration* [254]. In such a setting, the critical uncertainty from a user perspective is the uncertainty resulting in undercharging, i.e. the EV leaves earlier than expected due to an overestimated parking duration. As a result, a customised representation of uncertainty is required that ensures that the parking duration is not overestimated. This is almost identical to the challenge faced in public transport headway forecasts and opposite to the challenge faced by bike-sharing demand forecasts. Therefore, although we only consider one smart charging application in the present chapter, the findings are transferable to other mobility applications.

The rest of the chapter is structured as follows. In Section 7.1, we introduce the EV smart charging use case before considering existing literature related to uncertainty quantification for our use case in Section 7.2. In Section 7.3, we present our methodology for customising the representation of uncertainty quantification in parking duration forecasts. We discuss the case study used to evaluate our methodology in Section 7.4 before reporting all results in Section 7.5. We analyse and discuss these results in Section 7.6 before concluding in Section 7.7.

7.1 The Electric Vehicle Smart Charging Use Case

Smart charging of EVs is particularly important because EVs are considered as a major factor in reaching important climate targets, especially when charged by a highly renewable energy mix [255]. However, coupling this energy mix with an increased share of EVs causes new strains on our electrical system and can lead to grid instabilities [256]. As a result, coordinated and intelligent charging approaches, so-called smart charging, of EVs are required [256]–[259]. These intelligent charging approaches involve integrating EVs into a smart Internet of Things (IoT) electrical grid, enabling bi-directional communication to manage power flow, and optimising charging schedules [260]. However, these smart charging approaches should not inconvenience the user by, e.g., resulting in extra charging stops due to insufficient state of charge or forcing the user to charge at an unknown destination. Therefore, smart charging can only be successfully applied if information regarding a user's mobility behaviour is available and combined into the smart charging application [261]. This mobility behaviour includes common destinations, travel frequency, distance travelled, and how long a user stays at a specific location [261]. Additionally, for smart charging to be fully accepted, this mobility behaviour should be integrated into the smart charging application without the user manually feeding parameters into the SC algorithm [259], [262]. As a result, such mobility behaviour must be automatically predicted.

As with the mobility applications previously discussed, EV mobility also contains aspects of randomness [263]. For example, a EV user may leave for work every morning at a regular time, but due to fluctuating traffic conditions or unforeseen vehicle problems, the trip duration varies [264], [265]. Similarly, fluctuations in parking duration may be caused by external factors, such as a varying meeting schedule or after-work commitments [264]. The amplitude of these fluctuations depends on the individual EV user and their typical mobility habits [263]. Furthermore, this individuality also extends to a user's risk preference, e.g., some users may be willing to sacrifice a fully charged EV for a flexible schedule that maximises profit [266].

As a result, any forecast of a user's mobility behaviour must quantify this uncertainty, account for a user's individual risk preferences, and integrate this information into the smart charging algorithm. More importantly, the representation of this uncertainty should specifically account for the fact that undercharging, i.e. the EV leaves earlier than expected, is more problematic than an EV that is fully charged earlier than required. Such a representation of the uncertainty quantification allows for the application of stochastic smart charging algorithms [254], [259], [267], [268].

7.2 Related Work

Overall, few researchers have focused on parking duration forecast for a single user [189]. Instead, most research has forecast the demand of an EV either at a charging station, parking lot, or for a fleet of vehicles [269]–[273]. Furthermore, when considering probabilistic forecasts, almost all work focuses on electric vehicle charging demand [274]–[280], and does not consider the associated user-specific parking duration.

Machine learning methods are used to forecast both arrival and departure time in [254]. However, the paper's main focus is on the effects of this forecast on the scheduling and not on the accuracy of the forecast itself. Furthermore, Frendo *et al.* [254] focus on departure time point forecasts for a single location (a workplace) and a fleet of vehicles without quantifying the associated uncertainty. Uncertainty in parking duration and energy demand is considered in [189] via quantile forecasts for both quantities. These forecasts, however, are only performed for a single location, i.e., the home location [189]. Furthermore, Huber *et al.* [189] do not consider how to represent the uncertainty quantification in a way that is beneficial for smart charging. The first daily departure time is forecast in [281], and whilst prediction intervals based on an assumed Gaussian distribution of the errors are generated, the representation of this uncertainty for smart charging is again not considered. Mobility forecast for many vehicles is considered in [282] to analyse effects on a distribution grid, however, only point forecasts are considered. A review of scheduling, forecasting, and clustering strategies for EV charging is provided in [257], focusing on typical scheduling problems and coordinating the charging of multiple vehicles. Whilst probabilistic methods are discussed in [257], they again focus on EV charging demand or EV charging scheduling and do not consider the individual user's parking duration. Further mobility point forecasts are considered for a fleet of vehicles in [283] and in the form of next-place forecast in [284]–[286].

As shown in Table 7.1, none of the above papers specifically consider the representation of the quantified uncertainty in parking duration forecasts for user-centric smart charging. Most papers only consider point forecasts and only focus on a single location, or the locations are pre-labelled and not provided as GPS coordinates. When uncertainty is included, it is limited to a single location and only used to compare different forecasts, and the representation of this uncertainty is not customised specifically for smart charging. Therefore, in the remainder of this chapter, we focus on representations of the quantified uncertainty that are beneficial for smart charging.

Table 7.1.: Overview of related work that considers predicting a user's mobility behaviour. None of the identified papers focuses on representing the uncertainty quantification of parking duration forecasts specifically for smart charging. [5]

Paper	Quantity	Individual User	Multiple Locations	Probabilistic	Representing UQ
[189]	Parking Duration & Energy Demand	✓	✗	✓	✗
[254]	Parking Duration	✓	✗	✗	✗
[281]	First Daily Departure Time	✓	✗	✓	✗
[257]	Energy Demand & Schedule	✗	✓	✓	✗
[282]	Energy Demand	✗	✓	✗	✗
[283]	Energy Demand	✗	✗	✗	✗
[274]–[280]	Energy Demand	✗	✗	✓	✗
[269]–[273]	Energy Demand	✗	✗	✗	✗
[284]–[286]	Next-Place	✗	✓	✗	✗

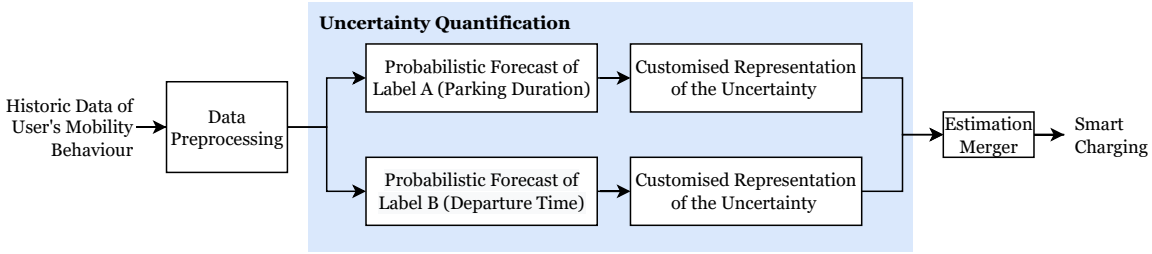


Figure 7.1.: Overview of our methodology with the focus of this chapter highlighted in blue. First, we preprocess data to derive known and commonly visited locations and create two forecast labels, parking duration (Label A) and departure time (Label B). Second, we generate a probabilistic parking duration forecast. Third, we create a customised representation of this uncertainty quantification designed for smart charging applications. Finally, the forecasts for both labels can be merged and integrated into stochastic smart charging. [5]

7.3 Methodology

Our methodology to customise the representation of uncertainty quantification specifically for smart charging applications is shown in Figure 7.1. First, the data is preprocessed before the uncertainty is quantified with probabilistic forecasts. In the third step, we customise the representation of this uncertainty quantification specifically for smart charging applications. The final step, which is not dealt with in the present chapter, is merging probabilistic forecasts for both labels and integrating this uncertainty into the smart charging application. In this section, we describe each of the first three steps.

7.3.1 Data Preprocessing

The data preprocessing includes three steps: data cleaning, spatial clustering, and data engineering. In the following, we describe these steps and provide an overview of the associated hyperparameters in Table 7.2.

Table 7.2.: An overview of the hyperparameters for the data preprocessing. [5]

Parameter	Value
Minimum Parking Time	2 h
Maximum Parking Time	24 h
Cluster Density Parameter (DBSCAN)	100 m
Minimum Number of Data Points per Cluster (DBSCAN)	5
Maximum Cluster Distance for Joining Clusters	500 m
Neighbourhood Radius to Assign Noise	300 m

Data Cleaning The first step in the preprocessing chain is data cleaning. The data cleaning initially involves removing all trips with measurement errors, for example, trips with invalid GPS locations or corrupt time stamps. After removing invalid trips, we calculate the parking duration, i.e. how long the vehicle is stationary before the next trip begins. We then remove all trips shorter than 15 s since we assume these to be measurement errors and unrealistic trip times. Finally, we aim to focus on parking durations relevant for smart charging applications. For a parking time of less than 2 h, there is not enough flexibility to enable smart charging. On the other hand, if the duration exceeds 24 h, there is too much flexibility, meaning smart charging is trivial. Therefore, we filter the data to only include parking durations between 2 h-24 h.

Spatial Clustering Given clean data, the next aspect of preprocessing is spatial clustering, described in more detail in [6]. Spatial clustering is necessary to determine key locations and account for small fluctuations in GPS coordinates. These fluctuations can occur when the destination is the same, but the exact parking location is slightly different, i.e. a different parking spot at the same supermarket. The spatial clustering consists of two steps. In the first step, we apply standard *DBSCAN*[1] [288] to the GPS parking location of all trips. Although this initial clustering generates several suitable clusters, it creates multiple clusters less than 500 m apart. Therefore, in the second spatially clustering step, we join clusters whose centroids are less than this predefined threshold of 500 m apart. Such clusters count as a single location for our charging purpose since a user would either walk between them (without using the EV) or, if they choose to drive, the energy consumption is negligible. Once these clusters are determined, we consider the *noise* points that do not belong to any location cluster. We assign a location labelled as noise to a given cluster if they are within a predefined neighbourhood radius of 300 m. We assume this neighbourhood radius is a reasonable distance for users to walk when parking at a known location. Finally, we label the clusters according to the frequency of their occurrence, i.e. the cluster that contains the most data points is “cluster 1”, that with the second-most data points “cluster 2”, and so on. We also label the noise data points with “-1”.

In the present chapter, we only consider the eight most frequently visited locations for both the training and evaluation of the parking duration forecasts. All trips with unknown end locations, i.e. those trips assigned to the *noise* cluster, are removed. This decision is made because a smart charging application is not possible if the location, and as a result, the charging infrastructure available is unknown.

[1] We apply the clustering algorithm with *Scikit-Learn* [287].

Table 7.3.: Additional features generated for the parking duration forecast. [5]

Feature	Description
Current Location	Cluster label for the current location of the EV given as a number.
Hour of Day	Time since midnight given as a real numeric value, e.g., 09:36 am is encoded as 9.6.
Time Window of Day	One-hot encoded variable to indicate different time windows of a day, i.e. morning (5-9 am), noon (9 am-1 pm), afternoon (1-5 pm), evening (5-10 pm), and night (10 pm-5 am).
Day of Week	Encoding of the day of the week (numeric value 0-6) to represent frequently recurring weekly trips.
Month	Month of the year (numeric value 1-12) to represent seasonally varying mobility behaviour.
Is Holiday	Boolean value based on a holiday calendar indicating which days are public and/or school holidays.
Last Parking Time at Current Location	How long the user stayed at the given location the last time they visited.

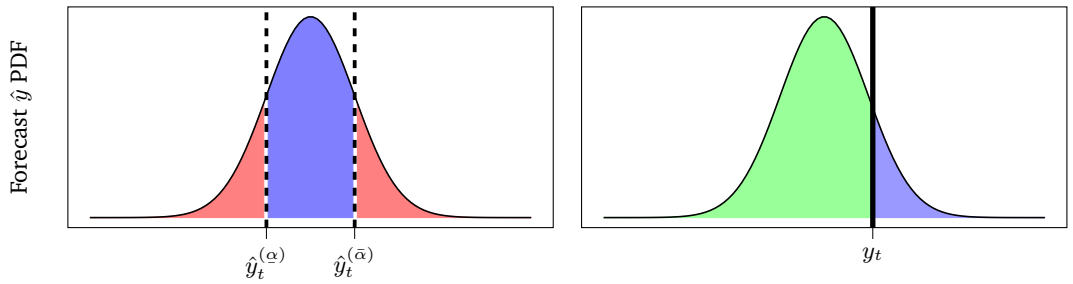
Data Engineering We engineer specific features from the end time of each trip, which are designed to provide useful information for the parking duration forecast. These additional features are shown and explained in Tab. 7.3. We also generate two labels for the parking duration forecasts. The first label is **Label A (Parking Duration)**, which is the time delta between arriving at the current location and departing for the next destination. The second label is **Label B (Departure Time)**, which is the point in time at which the next departure will occur, given the current location.

7.3.2 Uncertainty Quantification

The uncertainty quantification in our methodology consists of two steps. First we quantify the uncertainty with probabilistic forecasts. Second, given these probabilistic forecasts we customise the representation of the uncertainty so that it is beneficial for smart charging.

Probabilistic Forecasts

To quantify the uncertainty in parking duration forecasts, we generate probabilistic forecasts. For the present chapter, we focus on generating quantile forecasts, as defined in Section 2.3 via Equation (2.8). Furthermore, based on these quantile forecasts, we create prediction interval forecasts, defined in Equation (2.10). Thereby, our methodology is not limited to a given probabilistic forecasting model or method. Both parametric approaches, where a probabilistic distribution is assumed, and non-parametric distributions can be used. The difference is that in a parametric approach, the quantiles and Prediction Intervals (PIs) are calculated based on a forecast distribution, whilst in non-parametric approaches, these quantiles are forecast directly.



(a) Error Outside the Prediction Interval (b) Critical and Non-Critical Errors

Figure 7.2.: A schematic representation of two options to customise the representation of uncertainty for parking duration forecasts specifically for smart charging applications. The error outside the prediction interval shown in red in (a), assumes that the smart charging application is only at a disadvantage if the EV departs at a time outside the given prediction interval and thus only penalises observations outside this interval (see Equation (7.1)). Critical and non-critical error decomposition, shown in (b), makes a further distinction between forecasts that overestimate the parking duration resulting in the EV leaving earlier than expected and possibly undercharged (critical errors, shown in blue) and forecasts that underestimate the parking duration (non-critical errors, shown in green). This error decomposition is introduced in Equation (7.4). [5]

Customised Representation of the Uncertainty

Whilst the above-introduced forecasts quantify the uncertainty of parking duration forecasts, they fail to account for regions of uncertainty that may be critical for user-centric smart charging applications. Therefore, we now introduce three options to customise the representation of this uncertainty quantification specifically for smart charging, namely the *error outside the prediction interval*, an *error decomposition*, and *security levels*. We describe each of these options in the following.

Error Outside the Prediction Interval The first customised representation of uncertainty is created by comparing the error outside of the prediction interval and the width of that prediction interval. The idea behind this quantification is shown in Figure 7.2 (a). More precisely, we define the error outside of the prediction interval $\mathcal{E}_{t,\mathrm{PI}}$ as

$$\mathcal{E}_{t,\mathrm{PI}} = \begin{cases} \hat{y}_t^{(\underline{\alpha})} - y_t, & \text{if } y_t < \hat{y}_t^{(\underline{\alpha})}, \\ 0, & \text{if } \hat{y}_t^{(\underline{\alpha})} \leq y_t \leq \hat{y}_t^{(\bar{\alpha})}, \\ y_t - \hat{y}_t^{(\bar{\alpha})}, & \text{if } \hat{y}_t^{(\bar{\alpha})} < y_t, \end{cases} \tag{7.1}$$

for trip t, and the upper and lower quantile predictions for that trip $\hat{y}_t^{(\bar{\alpha})}$ and $\hat{y}_t^{(\underline{\alpha})}$, respectively. This quantification assumes that the only errors relevant for smart charging applications are those outside a given prediction interval, i.e., the red sections in Figure 7.2 (a). Therefore, the smart charging application is only at a disadvantage if the EV departs at a time that is not included in

the predicted interval. The $\mathcal{E}_{t,\text{PI}}$ is particularly useful when combined with the average width $\mathcal{W}$ of the prediction interval, defined as

$$\mathcal{W} = \frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t^{(\bar{\alpha})} - \hat{y}_t^{(\alpha)}), \tag{7.2}$$

for N considered trips. By jointly considering $\mathcal{E}_{\text{PI}}$ and the width of the prediction interval, a smart charging application can manage the trade-off between possible parking durations and errors.

Error Decomposition Whilst the combination of the error outside the prediction interval and the width of the prediction interval is useful for smart charging applications, it assumes that errors on both sides of the prediction interval are equally important. However, in the case of a user-centric smart charging application, forecasts that overestimate the parking duration, i.e., they predict the EV will depart later than it actually does are far more problematic. Such forecasts could lead to an incomplete charging cycle and an EV that cannot reach its destination without additional charging stops.

Therefore, we define the critical and non-critical errors for a user-centric smart charging application, shown in Figure 7.2 (b). First, in the deterministic case, we consider the total mean error for a parking duration forecast as the mean absolute error between the forecast and actual parking duration

$$\mathcal{E}_{\text{Tot}} = \frac{1}{N}\sum_{t=1}^{N}|\hat{y}_t - y_t|, \tag{7.3}$$

for each considered trip $t = 1, \ldots, N$. With this definition, we define the critical and non-critical components by rewriting Equation (7.3) as

$$\begin{aligned}\mathcal{E}_{\text{Tot}} &= \frac{1}{N}\sum_{t=1}^{N}\left(\mathbb{1}_{[\hat{y}_t \geq y_t]}\cdot(\hat{y}_t - y_t) + \mathbb{1}_{[\hat{y}_t < y_t]}\cdot(y_t - \hat{y}_t)\right)\\ &= \underbrace{\frac{1}{N}\sum_{t=1}^{N}\mathbb{1}_{[\hat{y}_t \geq y_t]}\cdot(\hat{y}_t - y_t)}_{\mathcal{E}_{\text{c}}} + \underbrace{\frac{1}{N}\sum_{t=1}^{N}\mathbb{1}_{[\hat{y}_t < y_t]}\cdot(y_t - \hat{y}_t)}_{\mathcal{E}_{\text{nc}}}\\ &= \mathcal{E}_{\text{c}} + \mathcal{E}_{\text{nc}},\end{aligned} \tag{7.4}$$

where $\mathcal{E}_{\text{c}}$ is the critical error for the parking duration forecast, $\mathcal{E}_{\text{nc}}$ the non-critical error, and $\mathbb{1}$ the indicator function. With this definition, we can create a representation of the uncertainty quantification that can be used to minimise critical errors when combined with smart charging algorithms. Formally, for a given tolerance $\mathcal{T}$ we can solve the optimisation problem

$$\begin{aligned}&\underset{\hat{y}_t}{\text{minimise}} \quad \mathbb{P}[\hat{y}_t > y_t \mid g, X, \hat{\Theta}]\\ &\text{subject to} \quad \mathcal{E}_{\text{nc}} = \frac{1}{N}\sum_{t=1}^{N}\mathbb{1}_{[\hat{y}_t < y_t]}\cdot(y_t - \hat{y}_t) \leq \mathcal{T},\end{aligned} \tag{7.5}$$

i.e., we aim to minimise the probability of a critical error occurring whilst ensuring that the total non-critical error is under a given threshold $\mathcal{T}$. This threshold is highly dependent on a user's individual risk preferences, and therefore, a method is required to determine appropriate thresholds.

Security Levels To help identify an appropriate threshold, we define quantile-based *security levels* that can be used to minimise the critical error depending on an individual EV user's risk preferences. Per definition, a quantile forecast $\hat{y}^{(\alpha)}$ ensures that the probability of the observation y being smaller than the quantile forecast is α. For example, a quantile forecast with $\alpha = 0.9$ should be larger than the observation 90% of the time. Regarding parking duration forecasts, this concrete example would also result in a critical error 90% of the time. Therefore, there is a clear mathematical relationship between quantile forecasts and the chance of a critical error. To take advantage of this relationship, we define security levels (SL) at level η,

$$\eta = 100 \cdot (1 - \alpha), \tag{7.6}$$

which minimise the critical error $\mathcal{E}_\mathrm{c}$ with increasing η. With this definition, a security level of η should guarantee that for $\eta\%$ of the observations, only non-critical errors occur. Whilst security levels do not determine a user-specific threshold $\mathcal{T}$, they provide a general starting point that can be used to determine approximate thresholds given a user's risk preferences.

7.4 Case Study

We evaluate our methodology to customise the representation of uncertainty in parking duration forecasts specifically for smart charging applications on two data sets and with four probabilistic forecast models. In this section, we introduce these data sets and forecast models before explaining the evaluation metrics applied.

7.4.1 Data

To evaluate the proposed approach, we consider two data sets: an openly available semi-synthetic data set with reduced uncertainty introduced in [6] and a real data set based on two years of real mobility behaviour that contains the full uncertainty of an EV user [259][2].

Semi-Synthetic Data As described in [6], we generate a semi-synthetic data set to create data representing a typical and predictable EV user. The semi-synthetic data set aims to replicate real user behaviour, excluding unpredictable events that cannot be accounted for in the optimisation process, for example, randomly visiting an unknown location. The semi-synthetic data set thus

[2]Due to data protection and privacy we cannot release the real data set openly.

contains reduced uncertainty compared to the real data. To achieve this goal, the semi-synthetic data set contains eight locations representing a real EV user's most commonly visited places. Furthermore, this semi-synthetic data set has no location "noise", as we assume only known locations are visited. To generate the semi-synthetic data set, we take real travel times between locations from a routing service and multiply them with a normally distributed random factor $k \sim \mathcal{N}(1, 0.05)$ to account for stochastic fluctuations in travel times. Given these trip times, we generate semi-synthetic sequences of trips, including time and location scatter with normally distributed offsets, to replicate the temporal and spatial variation in the trips. Furthermore, the trip sequences include recurrences on four levels: daily, weekly, monthly, and seasonally and two random trips per week to the grocery store occurring with a probability of 50% each. As a result, the semi-synthetic data set still includes uncertainty, but unexpected events or trips to unknown locations are removed.[3]

Real Data The real data is recorded from a personal vehicle over two years, from 2018 to 2019 [259]. During these two years, the vehicle was only used by a single person and was also the prioritised means of transport during this time frame [259]. The vehicle was equipped with an onboard computer to record trip data and parking duration. This data was then communicated via an IoT system (using the MQTT protocol) to a back-end database for storage (see also [260]). The final data set in the database consists of 2906 trips. Each trip includes GPS coordinates (which we use for spatial clustering) and timestamps (which we use to calculate the time-dependent features and labels). It is important to note that this real data set only mirrors the behaviour of one individual and is, therefore, not necessarily representative of other EV users. Furthermore, since this real data is collected from an individual using their EV to fulfil their mobility requirements, it contains all uncertainty associated with an EV user [259].

7.4.2 Probabilistic Forecast Models

We use probabilistic forecasts to quantify the uncertainty associated with parking duration predictions. When selecting probabilistic forecast models, we focus on robust models that are computationally inexpensive, openly available, and proven to perform well. Therefore, we exclude complex deep learning-based regression models that rely on extensive automated feature extraction, are computationally expensive to train, and are not openly available, e.g. [289]–[294]. Furthermore, since there is no clear correlation between successive trips, we also exclude time-series-based forecast models that consider auto-regressive terms, e.g. [40], [164], [170], [213]. Based on our selection criteria, we identify four probabilistic forecast models that are shown in Table 7.4: Bayesian Ridge Regression (BRR), Gaussian Process Regression (GPR), Natural Gradient Boosting (NGBoost), and a quantile regression Feed-Forward Neural Network (NN). Additionally, for each of these models, we consider a location-dependent ensemble, shown to be beneficial by Schwenk *et al.* [6]. The following briefly describes the general idea behind these

[3] The exact algorithm used for trip generation is openly available via GitHub `https://github.com/KarlSchwenk/mobility-data-creator`.

Table 7.4.: Overview of the selected probabilistic forecast models. [5]

Model	Uncertainty	Distribution	Relationship
BRR	Bayesian Prior	Parametric	Linear
GPR	Covariance Function	Parametric	Non-Linear
NGBoost	Gradient Boosting	Parametric	Non-Linear
NN	Quantile Loss	Non-Parametric	Non-Linear

Table 7.5.: An overview of the used hyperparameters for the selected probabilistic forecast models. [5]

Model	Hyperparameters
BRR	**Prior Distribution:** Spherical Gaussian **Max Number of Iterations:** 300 **Convergence Tolerance:** 0.003 **Regularisation Parameters Noise:** $\alpha_1 = \alpha_2 = 0.000006$ **Regularisation Parameters Weights:** $\lambda_1 = \lambda_2 = 0.000006$
GPR	**Kernel:** Radial Basis Function Kernel **Kernel Length Scale:** 1.0 **Noise Regularisation:** $\alpha = 0.1$ **Number of Optimizer Restarts:** 10
NGBoost	**Base Regressor:** Random Forest **Number of Trees in Random Forest:** 100 **Random Forest Loss Metric:** Mean Squared Error **NGBoost Distribution:** Gaussian distribution **NGBoost Number of Boosting Iterations:** 10000 **NGBoost Number of Early Stopping Rounds:** 10 **NGBoost Learning Rate:** 0.01 **NGBoost Scoring Rule:** Logarithmic Scoring Rule
NN	**Network Type:** Fully-Connected Feed-Forward Neural Network **Number of Hidden Layers:** 2 **Hidden Layer Size:** 100 Neurons, 50 Neurons **Hidden Layer Activation Function:** Rectified Linear Units **Output Activation Function:** Linear **Epochs:** 200 **Loss Metric:** Pinball Loss **Optimizer:** Adam [166]

probabilistic forecast methods and how the location ensemble is created. We refer to the existing literature for detailed mathematical descriptions of the applied models and present an overview of the used hyperparameters in Table 7.5.

Bayesian Ridge Regression The simplest probabilistic forecast model we apply is BRR. BRR is a Bayesian statistics approach for linear regression that incorporates prior distributions over the model parameters to regularise the estimates [295], [296]. Assuming a linear relationship between the input features and the parking duration target, BRR uses a prior distribution over the coefficients to quantify uncertainty. Since this assumed prior is a parametric distribution, BRR is classified as a parametric forecast method. We implement the BRR in *Python* [297] using *Scikit-Learn* [287] and assume the prior distribution to be a spherical Gaussian. For detailed information regarding BRR, we refer to Tipping [298], and MacKay [299].

Gaussian Process Regression Another simple probabilistic forecast model is GPR. The GPR is also based on Bayesian statistics, however, instead of assuming a specific distribution for the prior, it assumes a Gaussian process prior [300]. A Gaussian process is a collection of random variables of which any finite number has a joint Gaussian distribution [301]. As a result, GPR calculates the probability distribution over all admissible functions that fit the data and can, therefore model complex non-linear relationships. The considered Gaussian process is defined by a mean function and a covariance function, with uncertainty quantified with the covariance function. We implement the GPR in *Python* [297] using *Scikit-Learn* [287] with Gaussian process prior with a constant mean equal to that of the training data, and a radial basis function kernel with length-scale parameter equal to 1. For detailed information regarding GPR, we refer to Williams and Rasmussen [301].

Natural Gradient Boosting The third probabilistic forecast model is NGBoost, proposed by Duan et al. [177]. NGBoost applies gradient boosting [302] to optimise a probabilistic loss function. More specifically, NGBoost uses multi-parameter boosting and natural gradients to estimate the parameters of an assumed parametric probability distribution. NGBoost is based on an arbitrary deterministic base learner capable of modelling complex non-linear relationships [177]. In the training process, a separate base learner is trained for each parameter of the selected probability distribution using natural gradient boosting to minimise a proper scoring rule [177], such as the logarithmic score or continuous ranked probability score [32]. We implement NGBoost in *Python* [297], with the same random-forest base learner as in our previous work [6] using *Scikit-Learn* [287], and the *NGBoost* [177] Python package. We assume a Gaussian distribution [177] and apply the logarithmic proper scoring rule [32] for training. For detailed information regarding NGBoost we refer to Duan *et al.* [177] and Friedman [302].

Quantile Regression Neural Network The final probabilistic forecast model is a simple feed-forward quantile NN [171]. A quantile NN is trained like any feed-forward NN with gradient back-propagation, however, this training is designed to approximate a given target quantile directly. This approximation is achieved by training a NN with the Pinball Loss (PL) (see Equation (2.14)). In the present paper, we train multiple NNs to predict multiple quantiles and combine these predicted quantiles by sorting overlapping quantiles to achieve a non-parametric approximation of the full probability distribution. We implement the quantile feed-forward NN with two hidden layers of 100, and 50 neurons, respectively. The hidden layers use the rectified linear units (ReLU) activation function [303], whilst the output layer takes a linear activation function. Similar to [172], we predict 99 quantiles $\alpha \in \{0.01, \ldots, 0.99\}$, with individual NNs. The NN is implemented in *Python* [297] using *TensorFlow* [167] with *Keras* [168]. For detailed information on quantile neural networks, we refer to Koenker *et al.* [171], Gneiting *et al.* [15], and Goodfellow *et al.* [304].

Location Ensemble for Each Model To create a location ensemble, we first separate the training data into the known locations based on spatial clustering to quantify the uncertainty associated

with each location. Therefore, the *location ensemble* uses a separate model for each location to generate probabilistic parking duration forecasts. Although this allows the models to learn the varying levels of uncertainty at each location, it also leads to higher computational complexity, increasing with the number of known locations. Furthermore, the amount of training data available decreases when only a single location is considered, which may lead to an inaccurate representation of the uncertainty at this location. For each of the four probabilistic forecast models, we create a location ensemble.

7.4.3 Evaluation Metrics

We first consider a qualitative evaluation of the probabilistic forecasts by visualising prediction intervals. To further evaluate the probabilistic forecasts, we compare the predicted Cumulative Distribution Function (CDF) $\hat{F}(\hat{y})$, with the observed empirical CDF $F(y)$. To this means, we integrate over the absolute difference between the two CDFs, i.e.

$$\mathcal{E}_{\text{INT}} = \int_{2}^{24} |\hat{F}(z) - F(z)| dz, \tag{7.7}$$

where the integral between $2\,\text{h}$ and $24\,\text{h}$ is due to the filtered considered parking duration. In this case, a perfectly predicted CDF identical to the empirical CDF would result in an $\mathcal{E}_{\text{INT}}$ of zero, whilst the theoretical maximum $\mathcal{E}_{\text{INT}}$ is 22.

To evaluate our methodology for creating a customised representation of this uncertainty quantification, we consider the metrics defined in Section 7.3.2 and the mean error outside the prediction interval for all trips, i.e.,

$$\mathcal{E}_{\text{PI}} = \frac{1}{N} \sum_{t=1}^{N} \mathcal{E}_{t,\text{PI}}. \tag{7.8}$$

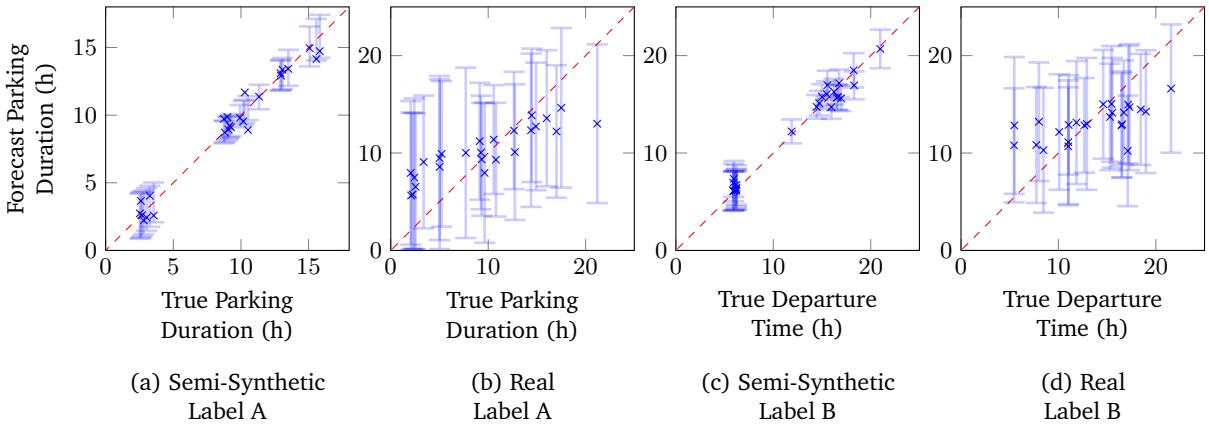
7.5 Results

In this section, we first analyse the probabilistic forecasts before reporting the results of our methodology for creating a customised representation of the uncertainty quantification specifically for smart charging. [4]

7.5.1 Probabilistic Forecasts

We first compare the prediction intervals before reporting differences between the predicted and empirical Probability Density Functions (PDFs) to analyse the uncertainty quantification from the probabilistic forecasts.

[4] Code to replicate the visualisations and error metrics considered is available via GitHub: `https://github.com/KIT-IAI/Customized-UQ-of-Parking-Duration-Predictions`.



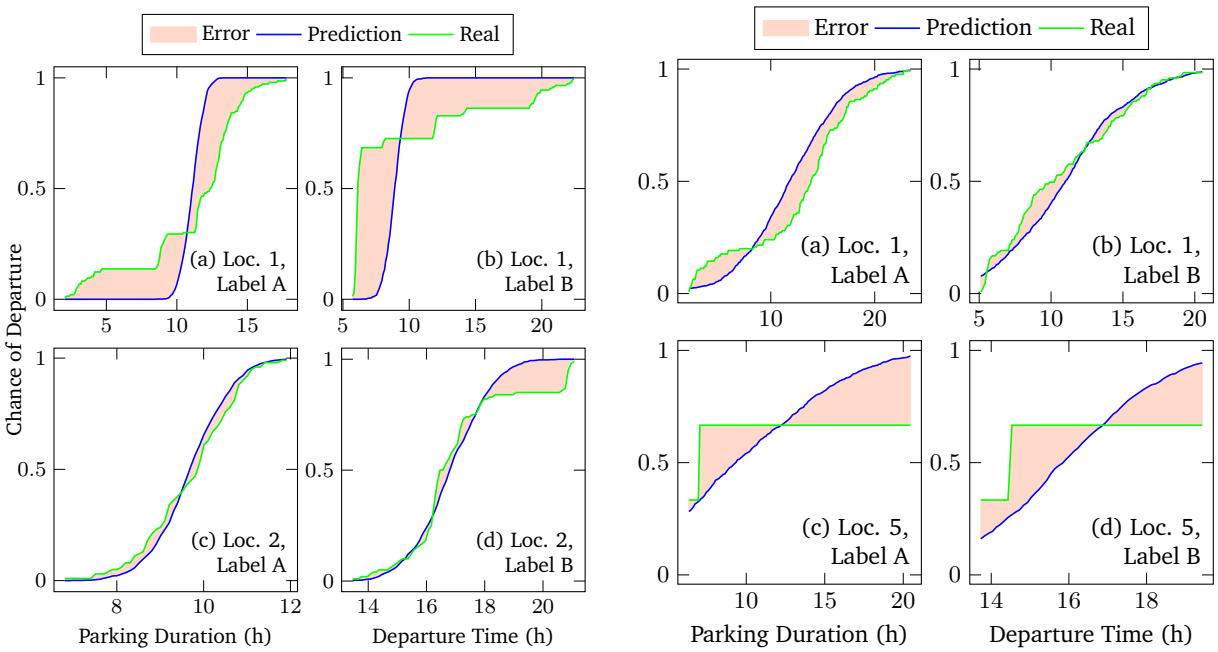
(a) Semi-Synthetic Label A (b) Real Label A (c) Semi-Synthetic Label B (d) Real Label B

Figure 7.3.: A comparison of the observed and forecast values from the NGBoost model for both Label A and B, on a subset of trips from the test data set. The red dotted line indicates a theoretical perfect point forecast, the mean point forecasts are blue crosses, and the light blue lines indicate the width of the 90% prediction interval. [5]

Prediction Intervals A comparison of the forecast values from NGBoost and observed values for 25 randomly selected trips from the test data set is shown in Figure 7.3. The blue crosses are the point forecasts obtained as the mean of the probabilistic forecast, and the 90% prediction intervals are shown by light blue bars. For both labels on the semi-synthetic data set, the mean forecasts always lie on or close to the diagonal, which indicates a perfect theoretical forecast. Furthermore, the prediction intervals for the semi-synthetic data are relatively narrow. Interestingly, prediction intervals for trips occurring in the middle of the considered time range, i.e. around $10\,\mathrm{h}$ for Label A and $12\,\mathrm{h}$ for Label B, are narrower than those for values at the edge of the considered time range. In contrast to the semi-synthetic data set, the mean forecasts for the real data set do not often lie on the desired diagonal. Furthermore, for both Labels A and B, the mean forecasts overestimate the parking duration for short stops and underestimate this duration for long stops. The prediction intervals on the real data set are also much wider than those from the semi-synthetic data set.

Probability Distribution To compare the true and predicted probabilistic distribution, we plot the predicted CDF and observed empirical CDF for two locations using NGBoost in Figure 7.4. For both data sets, the predicted CDF accuracy is highly dependent on the location. For example, on the semi-synthetic data set in Location 1, the predicted CDF is underdispersed and struggles to predict trips with either a very short or very long parking duration. On the other hand, the CDF for Location 2 on the semi-synthetic data set is highly accurate. For the real data, similar results are observed. For this data set, Location 1 results in an accurate CDF, whilst Location 5 is difficult to predict.

To further analyse the deviations in the predicted and true PDF for each location, we report the mean integral error, $\mathcal{E}_{\mathrm{INT}}$ for Label A and B in Table 7.6. Concerning the data sets, the errors are generally lower for the semi-synthetic data than the real data. However, for certain combinations of labels and locations, the errors are lower for the real data set. For both data sets, the location



(a) Semi-Synthetic **(b)** Real

Figure 7.4.: The predicted CDF and the observed CDF for the semi-synthetic data set (a) and real data set (b) calculated with the test data for two different locations using the NGBoost model. The error between the two distributions is the highlighted area between the two curves. [5]

plays a major role. Not only do the errors differ noticeably across the locations but also the best-performing label changes. For example, Label A is generally more accurate than Label B for the semi-synthetic data, but Label B delivers better results for Locations 3 and 7.

7.5.2 Customised Representation of the Uncertainty Quantification

In this section, we evaluate the customised representation of uncertainty by considering the error outside the prediction interval and the error decomposition results combined with different security levels.

Error Outside the Prediction Interval We report the trade-off between $\mathcal{E}_{\text{PI}}$ and $\mathcal{W}$ in Table 7.7 for Label A, and Table 7.8 for Label B. Comparing these tables, the errors on the semi-synthetic data set are lower than those from the real data, and at the same time, the width of the prediction intervals is also smaller. As expected, as the width of the prediction intervals increases, the error outside the prediction intervals decreases.

Considering Label A and the semi-synthetic data set, a minimum $\mathcal{E}_{\text{PI}}$ of $0.03\,\text{h}$ (approximately $1.8\,\text{min}$) is achieved with the BRR location ensemble with a $\mathcal{W}$ of $10.39\,\text{h}$. However, on this semi-synthetic data set, the NGBoost location ensemble performs similarly with a $\mathcal{E}_{\text{PI}}$ of $0.04\,\text{h}$ (approximately $2.4\,\text{min}$) and a lower $\mathcal{W}$ of $2.33\,\text{h}$. On the real data set, the smallest $\mathcal{E}_{\text{PI}}$ of $0.09\,\text{h}$ (approximately $5.4\,\text{min}$) is achieved with the GPR location ensemble, however, the $\mathcal{W}$ is $15.92\,\text{h}$.

Table 7.6.: The integral error, $\mathcal{E}_{\text{INT}}$, between the predicted distribution function and the actual observed distribution function for both labels. [5]

Data	Location	Label A				Label B			
		BRR	GPR	NGBoost	NN	BRR	GPR	NGBoost	NN
Semi-Synthetic Data	Mean for All Locations	0.9733	2.3763	2.4203	2.6101	2.0008	4.2096	4.2429	4.5634
	Location 1	1.4981	2.1349	2.3448	2.5258	2.7283	3.0856	3.5817	3.5413
	Location 2	0.2521	0.1421	0.1814	0.3618	0.5737	0.5012	0.4837	0.6468
	Location 3	1.8093	0.1752	0.1586	0.1663	0.0388	0.0375	0.0369	0.0896
	Location 4	0.2373	0.2627	0.2133	0.4149	0.2424	0.2378	0.2591	0.3413
	Location 5	1.6834	0.0925	0.0968	0.1167	1.4939	0.0468	0.0439	0.0540
	Location 6	5.6065	0.9957	0.1948	0.5113	7.2874	1.0292	0.2883	0.7282
	Location 7	0.4984	0.5314	0.5371	1.2980	0.4822	0.4953	0.4963	0.7570
	Location 8	0.5027	0.4848	0.5320	0.6274	2.0801	1.8908	1.8918	2.2633
Real Data	Mean for All Locations	0.9969	1.2119	1.4666	2.6141	0.9497	0.9855	1.1690	2.1500
	Location 1	1.4773	1.6453	1.8423	3.0827	0.6975	0.7479	0.7965	1.4152
	Location 2	0.6862	0.6840	0.8088	1.5589	0.9972	1.2688	1.2883	1.4998
	Location 3	1.1425	1.4316	1.1238	2.5132	1.3411	1.0643	1.2206	1.0875
	Location 4	0.1626	0.1603	0.1607	0.3090	1.7334	1.5058	2.0576	2.2980
	Location 5	3.8654	3.3779	3.3786	5.3612	3.1125	2.9198	1.2539	2.4804
	Location 6	2.2741	2.9024	3.0737	3.7693	1.6942	1.8172	2.2209	1.7147
	Location 7	0.8766	0.3412	0.2797	0.2812	0.7190	0.6626	0.9439	0.5664
	Location 8	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]

[a] There was only one trip for Location 8 in the test data set, making the calculation of an observed empirical CDF for this location impossible.

For Label B, the lowest $\mathcal{E}_{\text{PI}}$ of $0.07\,\text{h}$ (approximately $4.2\,\text{min}$) on the semi-synthetic data is achieved by NGBoost with a $\mathcal{W}$ of $3.67\,\text{h}$. For the real data, both BRR and the BRR location ensemble achieve the lowest $\mathcal{E}_{\text{PI}}$ of $0.07\,\text{h}$ (approximately $4.2\,\text{min}$), although the $\mathcal{W}$s of $13.92\,\text{h}$ and $13.77\,\text{h}$ respectively are far larger.

Error Decomposition & Security Levels The error decomposition in $\mathcal{E}_{\text{c}}$ and $\mathcal{E}_{\text{nc}}$ for security levels from 10% to 90% calculated with the NGBoost location ensemble is shown in Figure 7.5. Although the total error for the real data set is much larger than the semi-synthetic data, a high security level of 90% results in a similar small critical error. Furthermore, for both data sets, the minimal total error occurs at a security level between 40% to 60%.

To further analyse this error decomposition, we report the mean critical error $\mathcal{E}_{\text{c}}$, and non-critical error $\mathcal{E}_{\text{nc}}$ for both the semi-synthetic and real data set for Label A, and Label B in Table 7.9 and Table 7.10, respectively. Although the total error is much larger on the real data set, a high security level results in a small critical error for both sets. For example, for Label A, we can achieve an average critical error of only $0.03\,\text{h}$ (approximately $1.8\,\text{min}$) on the semi-synthetic data set and $0.1\,\text{h}$ (approximately $6\,\text{min}$) on the real data set, with a security level of 90%. Similarly, for Label B, the same security level can result in an average critical error of $0.01\,\text{h}$ (approximately $0.6\,\text{min}$) on the semi-synthetic data set and $0.09\,\text{h}$ (approximately $5.4\,\text{min}$) on the real data set. Comparing both labels, we observe that the error is not consistently lower for any one label but depends on the considered probabilistic forecast model.

Table 7.7.: The mean $\mathcal{E}_{\text{PI}}$ outside of the prediction interval and the average width $\mathcal{W}$ of this prediction interval in hours calculated on the test data for parking duration (Label A). All forecast models and their associated location ensemble (Ens) are compared. [5]

Data	Model	10%-PI		20%-PI		40%-PI		60%-PI		80%-PI		90%-PI	
		$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$
Semi-Synthetic Data	BRR	1.64	0.74	1.35	1.48	0.91	2.07	0.58	4.92	0.28	7.50	0.15	9.62
	GPR	0.61	0.19	0.54	0.38	0.41	0.78	030	1.25	0.20	1.91	0.14	2.45
	NGBoost	0.44	0.19	0.38	0.38	0.28	0.78	0.20	1.25	0.12	1.91	0.08	2.45
	NN	0.52	0.11	0.46	0.24	0.36	0.49	0.26	0.81	0.17	1.33	0.11	2.07
	BRR Ens	1.23	0.79	0.97	1.60	0.57	3.31	0.31	5.31	0.11	8.09	0.03	10.39
	GPR Ens	0.57	0.22	0.48	0.44	0.35	0.90	0.24	1.45	0.15	2.21	0.10	2.83
	NGBoost Ens	0.37	0.18	0.32	0.36	0.23	0.74	0.15	1.19	0.08	1.82	0.04	2.33
	NN Ens	0.51	0.11	0.47	0.20	0.39	0.41	0.31	0.67	0.20	1.20	0.13	1.88
Real Data	BRR	3.60	1.31	3.03	2.64	2.03	5.46	1.16	8.76	0.40	13.34	0.14	17.12
	GPR	3.84	1.29	3.28	2.59	2.20	5.36	1.46	8.60	0.67	13.09	0.36	16.80
	NGBoost	3.27	1.12	2.77	2.26	1.88	4.67	1.14	7.50	0.50	11.41	0.22	14.65
	NN	4.35	0.75	3.92	1.60	3.16	3.44	2.38	5.46	1.69	7.75	1.32	9.91
	BRR Ens	3.02	1.24	2.51	2.50	1.63	5.17	0.92	8.29	0.344	12.63	0.14	16.21
	GPR Ens	3.32	1.22	2.79	2.45	1.84	5.07	1.07	8.14	0.36	12.40	0.09	15.92
	NGBoost Ens	2.83	0.92	2.44	1.84	1.76	3.82	1.12	6.13	0.59	9.33	0.34	11.98
	NN Ens	5.02	0.54	4.76	1.08	4.27	2.21	3.65	3.93	2.73	6.21	2.27	7.94

7.6 Discussion

In this section, we first discuss the probabilistic forecasts before analysing our customised representation of this uncertainty quantification specifically designed for smart charging. Finally, we briefly discuss the implications of our findings for further mobility use cases.

Probabilistic Forecasts The first observation is that the increased uncertainty in the real data set is directly visible. When comparing the two data sets, the prediction intervals for the real data set are far wider than those for the semi-synthetic data set for all prediction models and labels. Interestingly, the mean integral error, $\mathcal{E}_{\text{INT}}$, is sometimes lower for the real data than the semi-synthetic data. This could be due to the extreme values observed in the real data. More specifically, whilst the predicted CDF is accurate for a majority of the values leading to a low $\mathcal{E}_{\text{INT}}$, it is extremely poor for the extreme values, which leads to the wide prediction intervals.

The second observation is that uncertainty is highly location-dependent. For certain locations, the predicted CDF is similar to the observed empirical CDF, whilst this prediction differs noticeably in other locations. Furthermore, both Label A and Label B can be beneficial depending on the location. This result is not surprising since a regular parking duration characterises some locations (e.g., visiting the gym), whilst other locations are characterised by a regular departure time (e.g., leaving work at the end of the working day).

Finally, the probabilistic forecasts emphasise the need for a customised representation of this uncertainty quantification for smart charging. More specifically, although a prediction interval or CDF is useful for visualising uncertainty, a smart charging algorithm will have difficulty optimising a charging schedule based purely on a large range of possible parking durations or an entire

Table 7.8.: The mean $\mathcal{E}_{\text{PI}}$ outside of the prediction interval and the average width $\mathcal{W}$ of this prediction interval in hours calculated on the test data for departure time (Label B). All forecast models and their associated location ensemble (Ens) are compared. [5]

Data	Model	10%-PI		20%-PI		40%-PI		60%-PI		80%-PI		90%-PI	
		$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$
Semi-Synthetic Data	BRR	2.55	1.09	2.14	2.20	1.44	4.54	0.82	7.29	0.31	11.11	0.14	14.26
	GPR	0.88	0.26	0.77	0.53	0.58	1.10	0.43	1.77	0.29	2.70	0.22	3.46
	NGBoost	0.59	0.28	0.51	0.56	0.37	1.17	0.25	1.87	0.13	2.85	0.07	3.67
	NN	0.64	0.15	0.57	0.32	0.44	0.65	0.31	1.09	0.21	1.79	0.14	3.07
	BRR Ens	1.83	0.98	1.43	1.98	0.86	4.09	0.52	6.57	0.27	10.00	0.12	12.84
	GPR Ens	0.80	0.32	0.68	0.63	0.50	1.29	0.37	2.08	0.23	3.16	0.15	4.06
	NGBoost Ens	0.67	0.2	0.60	0.40	0.47	0.83	0.36	1.34	0.24	2.03	0.18	2.61
	NN Ens	0.72	0.15	0.64	0.33	0.51	0.69	0.39	1.10	0.27	1.70	0.20	2.73
Real Data	BRR	2.61	1.06	2.16	2.14	1.37	4.44	0.72	7.12	0.25	10.84	0.07	13.92
	GPR	3.15	1.03	2.69	2.08	1.86	4.31	1.10	6.92	0.48	10.53	0.28	13.52
	NGBoost	2.64	0.93	2.24	1.88	1.51	3.89	0.86	6.24	0.29	9.50	0.10	12.19
	NN	3.97	0.62	3.62	1.30	2.92	2.75	2.35	4.19	1.80	5.91	1.44	7.51
	BRR Ens	2.49	1.05	2.04	2.12	1.26	4.39	0.62	7.05	0.19	10.73	0.07	13.77
	GPR Ens	2.69	0.99	2.25	1.99	1.43	4.12	0.74	6.61	0.22	10.07	0.11	12.92
	NGBoost Ens	2.61	0.90	2.21	1.81	1.43	3.74	0.77	6.01	0.27	9.14	0.14	11.74
	NN Ens	3.66	0.64	3.35	1.36	2.77	2.73	2.19	4.15	1.63	6.02	1.40	7.09

CDF. Furthermore, the low $\mathcal{E}_{\text{INT}}$ values for the real data, despite the large amounts of uncertainty, suggest that a complete CDF may also be misleading.

Customised Representation of the Uncertainty Quantification We first observe that quantifying the trade-off between the error outside the prediction interval and the prediction interval width only has limited benefit. To achieve small errors outside the prediction interval, we observe that large prediction intervals are required, which may not be viable in smart charging algorithms. Furthermore, the error outside the prediction interval does not explicitly quantify critical scenarios that may lead to undercharging.

Therefore, we observe that security levels based on probabilistic forecasts combined with an error decomposition provide the most useful quantification for smart charging applications. This customised representation of the uncertainty reduces the critical error to acceptable levels, even for real data exhibiting high uncertainty levels. Furthermore, using security levels can account for individual user's risk preferences and can be combined with a smart charging scheduling to optimise the smart charging application for that individual user. Given information regarding a user's risk attitude, it may be possible to optimally select a smart charging schedule that perfectly fits their profile.

However, we currently focus on user-centric smart charging applications, but the impact on other participants in an IoT smart grid should be considered. For example, an EV charging station owner who aims to schedule charging slots optimally will consider errors resulting in underestimated parking duration critical since the charging slot is not free for an additional EV when expected. Therefore, multiple definitions of critical errors, associated security levels, and the impacts on an IoT smart grid should be analysed.

Table 7.9.: The mean critical error $\mathcal{E}_c$ and non-critical error $\mathcal{E}_{nc}$ in hours on the test data set for different security levels (SL) for Label A. [5]

Data	Model	SL 10		SL 20		SL 30		SL 40		SL 50		SL 60		SL 70		SL 80		SL 90	
		$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$
Semi-Synthetic Data	BRR	3.95	0.08	2.85	0.26	2.09	0.44	1.52	0.65	1.05	0.93	0.70	1.31	0.47	1.88	0.31	2.64	0.20	3.80
	GPR	1.02	0.10	0.74	0.15	0.56	0.21	0.43	0.27	0.33	0.37	0.26	0.49	0.21	0.63	0.15	0.82	0.10	1.09
	NGBoost	0.99	0.06	0.70	0.11	0.51	0.15	0.36	0.20	0.25	0.28	0.18	0.40	0.13	0.56	0.10	0.75	0.06	1.04
	NN	0.63	0.12	0.47	0.17	0.38	0.23	0.31	0.27	0.25	0.33	0.19	0.39	0.13	0.48	0.09	0.60	0.04	0.86
	BRR Ens	4.01	0.01	2.67	0.05	1.80	0.18	1.19	0.42	0.76	0.79	0.55	1.38	0.40	2.04	0.26	2.82	0.11	3.87
	GPR Ens	1.18	0.05	0.85	0.10	0.63	0.16	0.47	0.22	0.35	0.32	0.26	0.45	0.19	0.61	0.13	0.83	0.09	1.17
	NGBoost Ens	0.91	0.04	0.64	0.08	0.46	0.13	0.32	0.18	0.21	0.25	0.14	0.36	0.10	0.51	0.06	0.70	0.03	0.98
	NN Ens	0.54	0.13	0.41	0.19	0.34	0.23	0.28	0.27	0.24	0.31	0.20	0.39	0.16	0.46	0.13	0.57	0.06	0.85
Real Data	BRR	6.99	0.27	5.02	0.58	3.75	0.96	2.82	1.45	2.14	2.09	1.58	2.85	1.07	3.74	0.58	4.90	0.13	6.72
	GPR	6.93	0.38	5.04	0.74	3.82	1.13	2.92	1.62	2.23	2.22	1.66	2.94	1.17	3.84	0.72	5.01	0.29	6.79
	NGBoost	6.24	0.33	4.57	0.62	3.46	0.92	2.65	1.33	2.01	1.80	1.45	2.37	0.96	3.09	0.53	4.07	0.17	5.65
	NN	4.08	1.31	3.65	1.52	3.14	1.79	2.68	2.08	2.26	2.47	1.84	2.84	1.37	3.45	0.86	4.20	0.38	5.36
	BRR Ens	5.98	0.24	4.12	0.55	2.95	0.94	2.13	1.46	1.51	2.09	1.05	2.88	0.69	3.85	0.37	5.08	0.10	6.93
	GPR Ens	5.63	0.16	3.83	0.50	2.74	0.94	2.05	1.56	1.58	2.32	1.22	3.19	0.90	4.17	0.57	5.38	0.20	7.14
	NGBoost Ens	4.30	0.40	3.01	0.71	2.24	1.10	1.67	1.51	1.25	2.02	0.93	2.61	0.66	3.34	0.41	4.24	0.20	5.63
	NN Ens	3.22	2.18	2.99	2.47	2.62	2.75	2.36	2.96	2.11	3.18	1.81	3.48	1.53	3.87	1.18	4.59	0.55	5.63

Table 7.10.: The mean critical error $\mathcal{E}_c$ and non-critical error $\mathcal{E}_{nc}$ in hours on the test data set for different security levels (SC), for Label B. [5]

Data	Model	SL 10		SL 20		SL 30		SL 40		SL 50		SL 60		SL 70		SL 80		SL 90	
		$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$	$\mathcal{E}_c$	$\mathcal{E}_{nc}$
Semi-Synthetic Data	BRR	6.11	0.30	4.47	0.57	3.34	0.81	2.42	1.06	1.65	1.40	1.07	1.92	0.63	2.65	0.25	3.63	0.01	5.26
	GPR	1.57	0.16	1.18	0.22	0.91	0.29	0.70	0.37	0.54	0.47	0.40	0.60	0.29	0.78	0.21	1.02	0.14	1.41
	NGBoost	1.58	0.05	1.15	0.11	0.86	0.17	0.61	0.23	0.41	0.30	0.28	0.46	0.20	0.68	0.14	0.97	0.08	1.40
	NN	0.84	0.17	0.66	0.22	0.52	0.27	0.40	0.33	0.31	0.40	0.24	0.48	0.17	0.57	0.09	0.75	0.04	1.16
	BRR Ens	5.16	0.24	3.67	0.47	2.62	0.66	1.79	0.89	1.11	1.19	0.54	1.61	0.20	2.29	0.05	3.34	0.02	4.90
	GPR Ens	1.61	0.18	1.16	0.26	0.85	0.34	0.60	0.43	0.40	0.54	0.25	0.71	0.16	0.95	0.10	1.29	0.05	1.78
	NGBoost Ens	1.06	0.15	0.77	0.21	0.58	0.28	0.44	0.35	0.33	0.44	0.25	0.56	0.19	0.72	0.14	0.92	0.10	1.22
	NN Ens	0.70	0.21	0.54	0.29	0.43	0.36	0.35	0.44	0.26	0.52	0.21	0.63	0.15	0.77	0.10	0.95	0.06	1.27
Real Data	BRR	5.82	0.13	4.16	0.34	2.09	0.61	2.30	0.97	1.69	1.43	1.19	2.00	0.76	2.71	0.39	3.68	0.11	5.27
	GPR	5.89	0.25	4.32	0.49	3.38	0.85	2.63	1.22	2.00	1.63	1.47	2.14	1.01	2.79	0.60	3.69	0.23	5.13
	NGBoost	5.18	0.13	3.78	0.36	2.90	0.65	2.23	0.99	1.69	1.39	1.25	1.89	0.86	2.50	0.50	3.31	0.16	4.61
	NN	3.60	1.38	3.21	1.58	2.83	1.76	2.42	2.01	2.06	2.23	1.60	2.50	1.16	2.84	0.78	3.33	0.41	4.11
	BRR Ens	5.44	0.08	3.84	0.32	2.83	0.64	2.08	1.03	1.49	1.50	1.01	2.07	0.62	2.82	0.31	3.83	0.11	5.48
	GPR Ens	5.22	0.13	3.75	0.40	2.84	0.73	2.16	1.12	1.61	1.56	1.14	2.08	0.70	2.71	0.34	3.60	0.09	5.07
	NGBoost Ens	4.85	0.16	3.51	0.39	2.71	0.72	2.10	1.08	1.58	1.47	1.12	1.91	0.72	2.47	0.38	3.26	0.12	4.57
	NN Ens	3.28	1.19	2.81	1.45	2.37	1.77	1.90	2.10	1.56	2.39	1.25	2.80	1.00	3.13	0.73	3.53	0.44	4.37

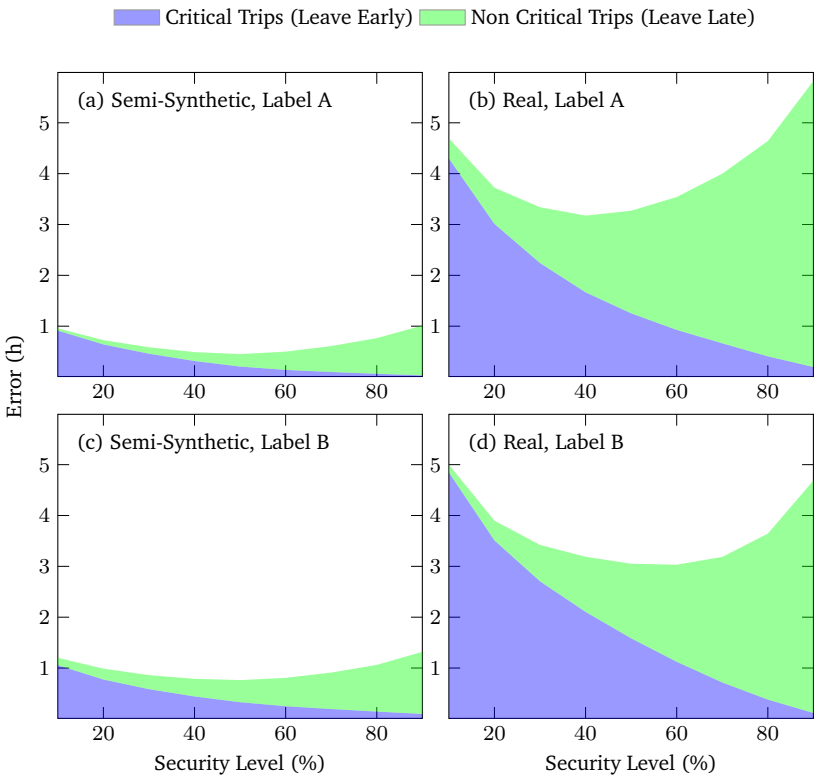


Figure 7.5.: The total mean error broken down into critical and non-critical error components for different security levels for the NGBoost ensemble calculated on the test data using all locations. The critical error decreases as the security level increases. [5]

Further Mobility Use Cases In this chapter, we only considered the EV smart charging use case, however, our methodology can be transferred to other mobility applications. For example, in public transport, headway forecasts critical errors also occur when the headway is overestimated and the public transport vehicle considered thus departs earlier than expected, causing passengers to miss their commute. Therefore, applying the same security levels and error decomposition will also benefit public transport headway forecasts. On the other hand, the problem of bike-sharing demand forecasts is inverse, i.e. the critical error occurs when underestimating peak demand. Thus, by simply reversing the definition of a security level to $\eta_{\text{Inverse}} = 100 \cdot \alpha$, it is possible to create security levels and an associated error decomposition that is suitable for the bike-sharing application. Therefore, although only validated on one use case, the customised representations discussed in this chapter should be helpful in many more mobility applications. Furthermore, if an asymmetrical cost function was provided that penalised over- or underestimation specifically for the given application, it would be possible to minimise these costs using α similar to the process undertaken in modern classification methods [305].

7.7 Conclusion

To enable a sustainable and efficient future mobility system, many mobility applications require uncertainty quantification in the form of probabilistic forecasts. However, these probabilistic

forecasts may not convey the uncertainty information in a way that is directly beneficial to that mobility application due to, for example, critical regions of uncertainty. Therefore, focusing on the use case of EV smart charging, this chapter introduces a methodology to create a customised representation of uncertainty quantification of parking duration forecasts designed to benefit smart charging. First, the uncertainty is quantified with probabilistic forecasts before a customised representation of this uncertainty quantification is created by decomposing critical errors that result in undercharging and non-critical errors. Furthermore, we define quantile-based security levels, which can minimise the probability of an EV being undercharged, given a user's risk preferences.

Using four probabilistic forecasting methods, we evaluate our approach on an openly available semi-synthetic data set and a real data set. We show that uncertainty is highly location-dependent and that probabilistic forecasts alone do not provide the specific information required by smart charging applications. However, our customised representation of this uncertainty quantification does provide such information by enabling critical errors to be reduced to acceptable levels for smart charging algorithms, even when high uncertainty exists in the data. Furthermore, due to the similarity of the smart charging use case with other mobility applications, our methodology can be transferred to further mobility applications with little or no modification.

In light of these findings, probabilistic forecasting models that automatically select the optimal label based on location-dependent uncertainty could be considered. Since the present paper focused on user-centric smart charging applications, future work should consider all participants in an IoT smart grid. Specifically, this work should investigate how the definition of critical errors varies for each participant and how these, perhaps contradictory, preferences can be combined to benefit all participants mutually. Furthermore, future work should focus on taking the customised uncertainty quantification presented in this paper and integrating it into stochastic optimisation problems, similar to [306], or using it to detect unusual behaviour, similar to [307]. Finally, the approach introduced in this chapter should be applied to further mobility use cases.

Part IV

Conclusion

8 Discussion

In the previous two parts, we dealt with quantifying and interpreting uncertainty. In Part II, we focused on quantifying uncertainty by presenting novel methods for generating probabilistic forecasts and customising the properties of these forecasts. In Part III, we focused on interpreting uncertainty, introducing an approach to explain the origins of uncertainty in a probabilistic forecast and also considering customised representations of uncertainty that highlight critical regions for mobility applications. Whilst each of the previous chapters included a discussion that addresses the key results and findings for that specific contribution, we have not yet discussed these individual contributions as a whole. Specifically, we have not considered how these individual contributions fit together to increase trust in time series forecasts. Therefore, in the present chapter, we address this question by discussing our findings and their impact on trust in forecasts in Section 8.1. Additionally, we discuss some main limitations and benefits of the work presented in the present dissertation in Section 8.2.

8.1 Findings and their Impact on Trust in Forecasts

In the present dissertation, we address five separate research questions and each of these research questions contributes to increasing the trust in a time series forecast. Therefore, in this section, we go through each of these research questions and highlight how the associated findings contribute to trust in forecasts.

RQ1 The first research question addressed whether it is possible to link the meteorological uncertainty to a time series forecast for a quantity that is affected by meteorological conditions. We show that, in principle, this is possible by propagating the uncertainty within a meteorological Ensemble Prediction System (EPS) through to a forecast for the target time series. However, since these EPSs are biased and underdispersed, ensemble post-processing is required. We show that post-processing in the final step is the most important, delivering forecasts with the best calibration and lowest Continuous Ranked Probability Score (CRPS). These findings offer both positive and negative implications for trust in the associated forecast. Regarding the positive implications, the approach generates probabilistic forecasts, and this quantification of uncertainty should immediately help increase trust in the forecast. Furthermore, the source of uncertainty used to generate the probabilistic forecast is known and tangible - i.e. the weather system. Compared to many probabilistic forecasting methods, which include uncertainty through specific loss functions or assumed distributions, this is an intuitive source of uncertainty that should also lead to the forecast being more tangible. However, the requirement for post-processing

decreases the trust. The applied post-processing method, Ensemble Model Output Statistics (EMOS), assumes a parametric distribution to perform calibration, meaning the meteorological uncertainty is only indirectly linked to the final probabilistic forecast. Furthermore, the simple fact that post-processing is required suggests that the meteorological uncertainty is not an ideal proxy for the uncertainty in the final forecast, which decreases trust. Finally, not all time series are affected by meteorological uncertainty, and therefore, the methods presented in Chapter 3 will only increase the trust in a specific class of time series forecasts. To summarise, the findings from Chapter 3 help to increase the trust in probabilistic forecasts by quantifying the uncertainty using a tangible source of uncertainty, however, they are only applicable to a small subset of all time series and limited by their use of parametric post-processing methods.

RQ2 In Chapter 4, we address the second research question, considering whether it is possible to use the underlying unknown data distribution of a given time series to include uncertainty in a forecast. We propose an approach that learns a mapping between the unknown data distribution and a known and tractable latent distribution with a Conditional Invertible Neural Network (cINN). This known and tractable latent space distribution can then be used to include uncertainty in an arbitrary point forecast. This approach also contributes to increasing trust in the forecast by quantifying the associated uncertainty, however, it is applicable to any time series and not just those affected by meteorological conditions. Furthermore, the source of the uncertainty included by our approach is in the data distribution itself. Although this uncertainty is not as tangible as meteorological uncertainty, it is still not entirely arbitrary and should, therefore, help increase trust in the forecast. However, our approach is a classic black-box machine learning approach. Whilst the idea itself is intuitive, the approach is not interpretable. For example, if the forecast is poor, it may not be clear if this results from a poor sampling hyperparameter or the cINN not accurately learning the mapping from the unknown distribution to the known and tractable latent space distribution. However, one advantage is that our approach includes uncertainty in existing arbitrary point forecasts. Therefore, if the underlying point forecast is trustworthy, then this should automatically increase trust in the resulting probabilistic forecast. To summarise, the contributions in Chapter 4 increase the trust in forecasts by quantifying the uncertainty for any time series, however, this forecast is not completely trustworthy due to the black-box nature of the approach.

RQ3 The third research question considers whether the properties of probabilistic forecasts can be customised without retraining the existing forecast model or developing an alternative model. In Chapter 5, we extend our approach that includes uncertainty in an arbitrary point forecast by integrated automated Hyperparameter Optimisation (HPO) and customising the forecasts to suit different probabilistic loss metrics. This approach goes one step further in increasing trust in the forecast than in the previous two contributions. Although the focus is still on quantifying uncertainty, our extension with HPO enables multiple probabilistic forecasts with customised properties to be easily generated. As a result, a probabilistic forecast can be designed to fulfil a user's requirements explicitly, and this added level of customisation should further

increase the trust in the forecast. Despite this increased trust, the resulting forecasts are not fully trustworthy. Our approach is still a black-box model and, therefore, difficult to interpret. Also, the customisation relies on automated HPO and customised probabilistic loss metrics, which is an added layer of machine learning and statistics. This extra layer may further decrease trust in the forecasts. Therefore, in summary, we observe that whilst the contribution from Chapter 5 should generate probabilistic forecasts that are more trustworthy than the previous contributions, they are still not truly trustworthy due to the added layers of machine learning and black-box nature of the approach.

RQ4 In Chapter 6, we consider whether the origins of uncertainty in a probabilistic forecast can be explained with existing Explainable Artificial Intelligence (XAI) methods. We propose an approach that separates the deterministic and uncertainty components of a probabilistic forecast within the network structure, enabling attribution-based XAI methods to explain the origins of uncertainty. Whilst an explanation of the origins of uncertainty will generally increase trust in a forecast, this may not always be the case. One reason is that communicating these explanations, often via visualisations, is not trivial for time series. As we observe in our evaluation in Chapter 6, it is not always trivial to interpret the visualisations and determine whether they were realistic. Without this intuitive representation of the explanations, it is difficult to trust the generated explanations, especially if the resulting probabilistic forecast appears realistic. We observe exactly this problem in Chapter 6, where the explanations for the real-world models seem to contradict the evaluation of the forecast quality. Therefore, whilst the contributions from Chapter 6 further increase trust in probabilistic forecasts, explaining the origins of the uncertainty is only truly useful if the explanations can be intuitively interpreted and confirm the findings from further evaluations.

RQ5 The fifth research question addresses how the quantified uncertainty can be represented in a manner that is beneficial for the downstream application. In Chapter 7, we consider mobility applications and highlight which regions of uncertainty are critical for smart charging. Given these critical regions, we propose how customised representations of the uncertainty can be used to minimise critical errors and convey useful information to the smart charging algorithm. Such customised representations of uncertainty should directly impact the trust associated with the forecast. Although this representation does not alter the forecast itself, it causes information that is important for the considered applications to be clearly highlighted. As a result, the information required to make decisions is presented in a helpful manner, automatically leading to a higher trust and acceptance of the forecast. However, since these customised representations do not alter the forecast, they are not useful if the forecast itself is poor. Therefore, whilst the contributions from Chapter 7 definitely increase the trust in the forecast, they are dependent on a high-quality forecast and are likely to be ineffective if the initial forecast does not contain any useful information.

Summary Although each individual contribution helps increase trust in forecasts, none alone is the solution to truly trustworthy forecasts. The true value of the present dissertation is the combination of these contributions. If a probabilistic forecast whose properties can be customised was generated, the origins of the uncertainty in this probabilistic forecast explained, and a useful representation of the uncertainty within this forecast created, then we would have a truly trustworthy forecast.

8.2 Limitations and Benefits

Whilst the present dissertation presented several novel contributions, it is also limited in some aspects. Therefore, in this section, we briefly highlight these key limitations, as well as mentioning some of the major benefits of the work.

Limitations The first key limitation of the present thesis is that it does not combine all contributions to create a truly trustworthy forecast. For example, meteorological uncertainty is only considered in Chapter 3, with EPSs not being considered in further chapters. Furthermore, the probabilistic forecasting methods using a cINN proposed in Chapter 4 and Chapter 5 cannot consider EPSs inputs without additional modification. Additionally, although the origins of uncertainty are explained in Chapter 6, this chapter does not consider the black-box models introduced in Chapter 4 and Chapter 5. Furthermore, the ability to customise probabilistic forecasts introduced in Chapter 5 is not applied to the mobility application considered in Chapter 7. As a result, although each of the contributions helps increase trust, based on the results of the present thesis, we cannot determine the effect of all contributions as a whole. Therefore, it would be interesting to investigate this aspect in more detail in future work by combining all contributions from the present dissertation to create a truly trustworthy forecast. Such a trustworthy forecast would quantify the uncertainty in a way customised for the considered application, consider exogenous sources of uncertainty, explain the origins of this uncertainty, and be represented in a useful way for the considered application.

Second, although many contributions are designed for real-world applications, we do not evaluate these applications in the field. This is particularly true for the loss-customised probabilistic forecasts proposed in Chapter 5 and the customised representations of uncertainty proposed in Chapter 7. Both of these contributions are only truly useful if they can be used to improve performance in a given application. If, for example, standard probabilistic forecasts perform just as well as forecasts customised for a selected application, then the loss customisation was not truly effective – even if it is able to alter the characteristics of the forecast. Similarly, if the customised representation of uncertainty does not enrich a smart charging application, then these representations also have little value. Therefore, it is imperative that further testing in a real-world setting is undertaken to validate and extend the results of this dissertation. A first step towards a real-world evaluation could be through the use of controlled research infrastructure, such as the Energy Lab 2.0 [308]. Such infrastructure that combines hardware and software to

enable controlled testing in close to real-world conditions could provide a platform to evaluate the present dissertation's contributions further. Furthermore, with applications such as smart charging [268] and battery storage optimisation [309] integrated into the infrastructure, the effects of real-world uncertainty on concrete applications can also be evaluated.

Finally, the present dissertation aims to quantify and interpret uncertainty to increase the trust in forecasts. However, we do not explicitly evaluate trust. Although trust can be estimated by considering different requirements, this is different than explicitly evaluating trust in the field. Therefore, it is necessary to analyse and evaluate trustworthiness in more detail to substantially determine how each of the contributions in the present dissertation contributes to trust. This step comprises multiple steps. First, based on discussions with experts and the users of forecasts, a formal definition of trustworthiness in time series forecasts and a list of criteria to measure this trustworthiness should be developed. Second, based on these criteria, a thorough survey could be undertaken to measure how trustworthy users find different types of probabilistic forecasts. Finally, this information should be used to adapt and improve the methods presented in this dissertation to improve the overall trust in the resulting probabilistic forecasts.

Benefits As well as the previously described limitations, the present dissertation has several benefits. First, we present multiple approaches to quantify the uncertainty in time series forecasts. These approaches are shown to generate reliable probabilistic forecasts that, in many cases, outperform existing benchmarks. Furthermore, we present an approach capable of generating forecasts with customised properties, which is useful for various applications.

Another key benefit of the present dissertation is that it takes the first steps towards explainable probabilistic forecasts. Whilst our approach for explaining the origins of uncertainty in probabilistic forecasts is simple, it is also the first of its kind. The valuable insights gained from these explanations should encourage further research in this field. In order for probabilistic forecasts to be successfully integrated into a variety of applications, it is imperative that the origins of uncertainty are clear and, therefore, such explanations will only grow in importance in the future.

Furthermore, the present dissertation also opens the door to a new research direction by focusing on the representation of probabilistic forecasts. Generally, forecasts in their pure form, i.e. quantiles or prediction intervals, are considered sufficient. However, we demonstrate that this is not always the case and that alternative representations that are designed for a specific application could be useful. Therefore, these observations should also encourage further research in this field and an investigation of how uncertainty can be best represented for a wide variety of applications.

Finally, all contributions in the present dissertation are made openly available via GitHub. These contributions include the base code for the presented methods and data and evaluation scripts to replicate the results presented here. By making our contributions openly available, we help to contribute to the open science movement and ensure our results are transparent and understandable for any interested party.

9 Summary and Outlook

With time series increasingly being integrated into a variety of applications in multiple domains, the demand for trustworthy forecasts of these time series is increasing. However, any forecast contains an inherent component of uncertainty, and for a forecast to be trustworthy, this uncertainty must be quantified with probabilistic forecasts. Although probabilistic forecasts help to increase trust, merely quantifying the uncertainty may not be sufficient for the forecast to be truly trustworthy. The quantified uncertainty must also be interpreted in a manner that is useful for the application using the probabilistic forecasts. Therefore, the present dissertation addresses the challenges of quantifying and interpreting uncertainty in time series forecasts. We answer five research questions to address these challenges, leading to five novel contributions.

The first novel contribution is a comparison of post-processing strategies designed to link meteorological uncertainty to time series that are affected by this uncertainty. We show that the meteorological uncertainty can be linked to a further time series, however, post-processing is required to generate high-quality and calibrated forecasts. Furthermore, post-processing the final target forecast is the most important step. The second contribution is a novel method to generate probabilistic forecasts from arbitrary point forecasts using a Conditional Invertible Neural Network (cINN). We show how this cINN can be employed to learn a mapping from the unknown distribution of the considered time series to a known and tractable distribution in the latent space. We then use this latent space distribution to include uncertainty in arbitrary point forecasts. The third contribution is an extension of this approach with automated Hyperparameter Optimisation (HPO), which enables the properties of the resulting probabilistic forecasts to be customised according to a given probabilistic loss metric. Importantly, this customisation is possible without retraining either the base forecasting model or the cINN. These first three contributions all deal with quantifying uncertainty to increase the trust in the forecast. Whilst the first two focus on merely generating probabilistic forecasts, the third contribution goes a step further by enabling the customised generation of probabilistic forecasts, which should further increase trust in this forecast.

The fourth and fifth contributions of the present dissertation deal with interpreting uncertainty. The fourth contribution is an approach that enables the origins of uncertainty in a probabilistic forecast to be explained with existing attribution-based Explainable Artificial Intelligence (XAI) methods. By separating the deterministic and uncertain components of the probabilistic forecast in the network architecture, we can apply existing XAI methods to explain both components, including the origins of uncertainty. We show that these explanations are plausible and deliver important insights when evaluated on real-world data. The fifth contribution focuses on representations of uncertainty that are beneficial for downstream applications. Focusing on mobility applications, we identify regions of uncertainty that are critical for smart charging applications.

Given these critical regions of uncertainty, we propose various representations of the quantified uncertainty that highlight critical regions so they can be better considered in smart charging applications. Although we only consider a single mobility application, our approach is easily extendable to other applications.

In light of these findings, there are numerous opportunities for future work. First, the individual contribution of this dissertation should be combined to create truly trustworthy forecasts. Such a forecast should consider exogenous sources of uncertainty, such as weather forecasts, and be capable of generating probabilistic forecasts whose properties can be customised depending on the requirements. Furthermore, this truly trustworthy forecast should be interpretable by explaining the origins of the quantified uncertainty and being represented in a beneficial way for the considered application. Second, the interpretable nature of probabilistic forecasts should be considered in more detail. Thereby, explanation methods that better account for the temporal dependencies in time series should be developed to explain the origins of uncertainty better, improved visualisation could be considered and, ideally, probabilistic forecasting models that are naturally interpretable should be created. Third, the results obtained in the present dissertation should be validated and extended for use in real-world scenarios. For example, probabilistic forecasts should be generated and customised to fulfil the requirements of a given downstream application, and the performance improvement of this application should be monitored. Alternatively, the customised representation of uncertainty could be integrated into stochastic smart charging algorithms and evaluated by real users. Finally, the present dissertation quantifies and interprets uncertainty to increase the trust in forecasts, however, we do not explicitly investigate what forecasts are considered to be trustworthy by an end user. Therefore, future work should investigate the definition of trustworthiness in more detail and explicitly evaluate how each contribution affects trust.

Bibliography

[1] K. Phipps, S. Lerch, M. Andersson, R. Mikut, V. Hagenmeyer, and N. Ludwig, "Evaluating ensemble post-processing for wind power forecasts", *Wind Energy*, vol. 25, no. 8, pp. 1379–1405, 2022. DOI: 10.1002/we.2736.

[2] K. Phipps, N. Ludwig, V. Hagenmeyer, and R. Mikut, "Potential of ensemble copula coupling for wind power forecasting", in *Proceedings 30. Workshop Computational Intelligence*, vol. 26, KIT Scientific Publishing, 2020, p. 87. DOI: 10.5445/IR/1000127955.

[3] K. Phipps, B. Heidrich, M. Turowski, M. Wittig, R. Mikut, and V. Hagenmeyer, "Generating probabilistic forecasts from arbitrary point forecasts using a conditional invertible neural network", *Applied Intelligence*, 2024. DOI: https://doi.org/10.1007/s10489-024-05346-9.

[4] K. Phipps, S. Meisenbacher, B. Heidrich, M. Turowski, R. Mikut, and V. Hagenmeyer, "Loss-customised probabilistic energy time series forecasts using automated hyperparameter optimisation", in *Proceedings of the Fourteenth ACM International Conference on Future Energy Systems*, ACM, 2023, pp. 271–286. DOI: 10.1145/3575813.3595204.

[5] K. Phipps, K. Schwenk, B. Briegel, R. Mikut, and V. Hagenmeyer, "Customized uncertainty quantification of parking duration predictions for EV smart charging", *IEEE Internet of Things Journal*, pp. 1–1, 2023. DOI: 10.1109/JIOT.2023.3299201.

[6] K. Schwenk, K. Phipps, B. Briegel, V. Hagenmeyer, and R. Mikut, "A benchmark for parking duration prediction of electric vehicles for smart charging applications", in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2021, pp. 1–8. DOI: 10.1109/SSCI50451.2021.9660063.

[7] B. Heidrich, A. Bartschat, M. Turowski, O. Neumann, K. Phipps, S. Meisenbacher, K. Schmieder, N. Ludwig, R. Mikut, and V. Hagenmeyer, "pyWATTS: Python workflow automation tool for time series", 2021. arXiv: 2106.10157.

[8] M. Turowski, B. Heidrich, K. Phipps, K. Schmieder, O. Neumann, R. Mikut, and V. Hagenmeyer, "Enhancing anomaly detection methods for energy time series using latent space data representations", in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ACM, 2022, pp. 208–227. DOI: 10.1145/3538637.3538851.

[9] S. Meisenbacher, M. Turowski, K. Phipps, M. Rätz, D. Müller, V. Hagenmeyer, and R. Mikut, "Review of automated time series forecasting pipelines", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, no. 6, e1475, 2022. DOI: 10.48550/arXiv.2202.01712.

[10] B. Heidrich, M. Turowski, K. Phipps, K. Schmieder, W. Süss, R. Mikut, and V. Hagenmeyer, "Controlling non-stationarity and periodicities in time series generation using conditional invertible neural networks", *Applied Intelligence*, pp. 1–18, 2022. DOI: 10.1007/s10489-022-03742-7.

[11] M. Turowski, M. Weber, O. Neumann, B. Heidrich, K. Phipps, H. K. Çakmak, R. Mikut, and V. Hagenmeyer, "Modeling and generating synthetic anomalies for energy and power time series", in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ACM, 2022, pp. 471–484. DOI: 10.1145/3538637.3539760.

[12] B. Heidrich, L. Mannsperger, M. Turowski, K. Phipps, B. Schäfer, R. Mikut, and V. Hagenmeyer, "Boost short-term load forecasts with synthetic data from transferred latent space information", in *Proceedings of the 11th DACH+ Conference on Energy Informatics*, vol. 5, SpringerOpen, 2022, pp. 1–20. DOI: 10.1186/s42162-022-00214-7.

[13] M. Beichter, K. Phipps, M. M. Frysztacki, R. Mikut, V. Hagenmeyer, and N. Ludwig, "Net load forecasting using different aggregation levels", in *Proceedings of the 11th DACH+ Conference on Energy Informatics*, vol. 5, SpringerOpen, 2022, pp. 1–21. DOI: 10.1186/s42162-022-00213-8.

[14] M. Luh, K. Phipps, A. Britto, M. Wolf, M. Lutz, and J. Kraft, "High-resolution real-world electricity data from three microgrids in the global south", in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, ACM, 2022, pp. 496–514. DOI: 10.1145/3538637.3539763.

[15] T. Gneiting, D. Wolffram, J. Resin, K. Kraus, J. Bracher, T. Dimitriadis, V. Hagenmeyer, A. I. Jordan, S. Lerch, K. Phipps, *et al.*, "Model diagnostics and forecast evaluation for quantiles", *Annual Review of Statistics and Its Application*, vol. 10, 2023. DOI: 10.1146/annurev-statistics-032921-020240.

[16] B. Heidrich, K. Phipps, O. Neumann, M. Turowski, R. Mikut, and V. Hagenmeyer, "ProbPNN: Enhancing deep probabilistic forecasting with statistical information", 2023. arXiv: 2302.02597.

[17] D. Werling, M. Beichter, B. Heidrich, K. Phipps, R. Mikut, and V. Hagenmeyer, "The impact of forecast characteristics on the forecast value for the dispatchable feeder", in *Companion Proceedings of the 14th ACM International Conference on Future Energy Systems*, ACM, 2023, pp. 59–71. DOI: 10.1145/3599733.3600251.

[18] R. Poppenborg, K. Phipps, H. Khalloof, K. Förderer, R. Mikut, and V. Hagenmeyer, "Dynamic chromosome interpretation in evolutionary algorithms for distributed energy resources scheduling", in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, ACM, 2023, pp. 755–758. DOI: 10.1145/3583133.3590666.

[19] D. Werling, M. Beichter, B. Heidrich, K. Phipps, R. Mikut, and V. Hagenmeyer, "Automating value-oriented forecast model selection by meta-learning: Application on a dispatchable feeder", in *Proceedings of the Energy Informatics.Academy Conference 2022 (EI.A 2022)*, in press, SpringerOpen, 2023.

[20] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and practice*, 2nd ed. Melbourne, Australia: OTexts, 2018.

[21] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*, 3rd ed. Cham, Switzerland: Springer, 2002.

[22] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, 2015.

[23] M. Turowski, "Data-driven methods for managing anomalies in energy time series", Ph.D. dissertation, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2023.

[24] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting", *International Journal of Forecasting*, vol. 22, no. 3, pp. 443–473, 2006. DOI: 10.1016/j.ijforecast.2006.01.001.

[25] M. S. Gönül, D. Önkal, and P. Goodwin, "Why should I trust your forecasts?", *Foresight: The International Journal of Applied Forecasting*, no. 27, pp. 5–9, 2012.

[26] J. N. Burgeno and S. L. Joslyn, "The impact of weather forecast inconsistency on user trust", *Weather, Climate, and Society*, vol. 12, no. 4, pp. 679–694, 2020. DOI: 10.1175/WCAS-D-19-0074.1.

[27] S. Joslyn and J. LeClerc, "Decisions with uncertainty: The glass half full", *Current Directions in Psychological Science*, vol. 22, no. 4, pp. 308–315, 2013. DOI: 10.1177/0963721413481473.

[28] F. Monforti and I. Gonzalez-Aparicio, "Comparing the impact of uncertainties on technical and meteorological parameters in wind power time series modelling in the European Union", *Applied Energy*, vol. 206, pp. 439–450, 2017. DOI: 10.1016/j.apenergy.2017.08.217.

[29] D. Seuss, "Bridging the gap between explainable AI and uncertainty quantification to enhance trustability", 2021. arXiv: 2105.11828.

[30] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges", *Information Fusion*, vol. 76, pp. 243–297, 2021. DOI: 10.1016/j.inffus.2021.05.008.

[31] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence", *Nature*, vol. 521, no. 7553, pp. 452–459, 2015. DOI: 10.1038/nature14541.

[32] T. Gneiting and M. Katzfuss, "Probabilistic forecasting", *Annual Review of Statistics and Its Application*, vol. 1, no. 1, pp. 125–151, 2014. DOI: 10.1146/annurev-statistics-062713-085831.

[33] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications", *IEEE Access*, vol. 6, pp. 36 218–36 234, 2018. DOI: 10.1109/ACCESS.2018.2836917.

[34] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey", *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20 200 209, 2021. DOI: 10.1098/rsta.2020.0209.

[35] J. P. Martino, "A review of selected recent advances in technological forecasting", *Technological Forecasting and Social Change*, vol. 70, no. 8, pp. 719–733, 2003. DOI: 10.1016/S0040-1625(02)00375-X.

[36] J. Nowotarski and R. Weron, "Recent advances in electricity price forecasting: A review of probabilistic forecasting", *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1548–1568, 2018. DOI: 10.1016/j.rser.2017.05.234.

[37] Y. Zhang, J. Wang, and X. Wang, "Review on probabilistic forecasting of wind power generation", *Renewable and Sustainable Energy Reviews*, vol. 32, pp. 255–270, 2014. DOI: 10.1016/j.rser.2014.01.033.

[38] L. Li, J. Yan, X. Yang, and Y. Jin, "Learning interpretable deep state space model for probabilistic time series forecasting", 2021. arXiv: 2102.00397.

[39] T. Kono, S. Yamaguchi, and T. Nagao, "Time series prediction with dual reliability: Uncertainty and explainability", in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2020, pp. 4095–4102. DOI: 10.1109/SMC42975.2020.9283170.

[40] B. Lim, S. Ö. Ark, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting", *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021. DOI: 10.1016/j.ijforecast.2021.03.012.

[41] F. Petropoulos, R. J. Hyndman, and C. Bergmeir, "Exploring the sources of uncertainty: Why does bagging for time series forecasting work?", *European Journal of Operational Research*, vol. 268, no. 2, pp. 545–554, 2018. DOI: 10.1016/j.ejor.2018.01.045.

[42] K. Wickstrøm, K. Ø. Mikalsen, M. Kampffmeyer, A. Revhaug, and R. Jenssen, "Uncertainty-aware deep ensembles for reliable and explainable predictions of clinical time series", *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 7, pp. 2435–2444, 2020. DOI: 10.1109/JBHI.2020.3042637.

[43] C. Zhao, C. Wan, and Y. Song, "Cost-oriented prediction intervals: On bridging the gap between forecasting and decision", *IEEE Transactions on Power Systems*, vol. 37, no. 4, pp. 3048–3062, 2021. DOI: 10.1109/TPWRS.2021.3128567.

[44] D. Leffrang and O. Müller, "Should I follow this model? The effect of uncertainty visualization on the acceptance of time series forecasts", in *IEEE Workshop on TRust and EXpertise in Visual Analytics (TREX)*, IEEE, 2021, pp. 20–26. DOI: 10.1109/TREX53765.2021.00009.

[45] J. M. Morales, A. J. Conejo, H. Madsen, P. Pinson, and M. Zugno, *Integrating Renewables in Electricity Markets*. New York, NY, USA: Springer, 2013.

[46] M. Sundararajan and A. Najmi, "The many Shapley values for model explanation", in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, PMLR, 2020, pp. 9269–9278. DOI: 10.48550/arXiv.1908.08474.

[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization", in *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74.

[48] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview", in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Cham, Switzerland: Springer, 2019, pp. 193–209.

[49] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Diaz-Rodriguez, "Explainable artificial intelligence (XAI) on timeseries data: A survey", 2021. arXiv: 2104.00950.

[50] R. Barbano, S. Arridge, B. Jin, and R. Tanno, "Uncertainty quantification in medical image synthesis", in *Biomedical Image Synthesis and Simulation*. London, UK: Elsevier, 2022, pp. 601–641.

[51] W. W. Wei, *Time series analysis*. Boston, MA, USA: Pearson Education, Inc., 2013.

[52] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: Forecasting and control*. Hoboken, NJ, USA: John Wiley & Sons, 2015.

[53] D. R. Cox and H. D. Miller, *The theory of stochastic processes*. Berlin, Germany: Springer, 1977, vol. 134.

[54] S. M. Ross, *Stochastic processes*. New York, NY, USA: John Wiley & Sons, 1995.

[55] J. Lamperti, *Stochastic processes: A survey of the mathematical theory*. New York, NY, USA: Springer, 2012, vol. 23.

[56] S. Asmussen, *Applied probability and queues*. New York, NY, USA: Springer, 2003, vol. 2.

[57] O. Kallenberg, *Foundations of modern probability*. Cham, Switzerland: Springer, 1997, vol. 2.

[58] D. W. Hubbard, *How to measure anything: Finding the value of intangibles in business*. Hoboken, NJ, USA: John Wiley & Sons, 2014.

[59] P. F. Pelz, M. E. Pfetsch, S. Kersting, M. Kohler, A. Matei, T. Melz, R. Platz, M. Schaeffner, S. Ulbrich, S. Kersting, *et al.*, "Types of uncertainty", in *Mastering Uncertainty in Mechanical Engineering*. Cham, Switzerland: Springer, 2021, pp. 25–42.

[60] F. Knight, *Risk, uncertainty and profit*. Washington, WA, USA: Beard Books, 2013.

[61] Z. Guo, Z. Wan, Q. Zhang, X. Zhao, F. Chen, J.-H. Cho, Q. Zhang, L. M. Kaplan, D. H. Jeong, and A. Jøsang, "A survey on uncertainty reasoning and quantification for decision making: Belief theory meets deep learning", 2022. arXiv: 2206.05675.

[62] A. Jsang, *Subjective Logic: A formalism for reasoning under uncertainty*. Cham, Switzerland: Springer, 2018.

[63] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? Does it matter?", *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009. DOI: 10.1016/j.strusafe.2008.06.020.

[64] I. Linkov and D. Burmistrov, "Model uncertainty and choices made by modelers: Lessons learned from the international atomic energy agency model intercomparisons", *Risk Analysis: An International Journal*, vol. 23, no. 6, pp. 1297–1308, 2003. DOI: 10.1111/j.0272-4332.2003.00402.x.

[65] W. E. Walker, P. Harremoës, J. Rotmans, J. P. Van Der Sluijs, M. B. Van Asselt, P. Janssen, and M. P. Krayer von Krauss, "Defining uncertainty: A conceptual basis for uncertainty management in model-based decision support", *Integrated Assessment*, vol. 4, no. 1, pp. 5–17, 2003. DOI: 10.1076/iaij.4.1.5.16466.

[66] M. Brugnach, A. Dewulf, C. Pahl-Wostl, and T. Taillieu, "Toward a relational concept of uncertainty: About knowing too little, knowing too differently, and accepting not to know", *Ecology and Society*, vol. 13, no. 2, 2008.

[67] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction", *Nature*, vol. 525, no. 7567, pp. 47–55, 2015. DOI: 10.1038/nature14956.

[68] H.-J. Zimmermann, "An application-oriented view of modeling uncertainty", *European Journal of Operational Research*, vol. 122, no. 2, pp. 190–198, 2000. DOI: 10.1016/S0377-2217(99)00228-3.

[69] A. Mitsos, N. Asprion, C. A. Floudas, M. Bortz, M. Baldea, D. Bonvin, A. Caspari, and P. Schäfer, "Challenges in process optimization for new feedstocks and energy sources", *Computers & Chemical Engineering*, vol. 113, pp. 209–221, 2018.

[70] S. Sass, A. Tsoukalas, I. H. Bell, D. Bongartz, J. Najman, and A. Mitsos, "Towards global parameter estimation exploiting reduced data sets", *Optimization Methods and Software*, pp. 1–13, 2023.

[71] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 2, pp. 243–268, 2007. DOI: 10.1111/j.1467-9868.2007.00587.x.

[72] 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", *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 10, no. 6, pp. 497–516, 2007. DOI: 10.1002/we.230.

[73] B. Schulz and S. Lerch, "Machine learning methods for postprocessing ensemble forecasts of wind gusts: A systematic comparison", *Monthly Weather Review*, vol. 150, no. 1, pp. 235–257, 2022. DOI: 10.1175/MWR-D-21-0150.1.

[74] Z. Toth, O. Talagrand, G. Candille, and Y. Zhu, "Probability and ensemble forecasts", in *Forecast Verification: A Practitioner's Guide in Atmospheric Science*. Chichester, England: John Wiley and Sons, 2003, vol. 137, p. 163.

[75] M. S. Roulston and L. A. Smith, "Evaluating probabilistic forecasts using information theory", *Monthly Weather Review*, vol. 130, no. 6, pp. 1653–1660, 2002. DOI: 10.1175/1520-0493(2002)130<1653:EPFUIT>2.0.CO;2.

[76] J. L. Anderson, "A method for producing and evaluating probabilistic forecasts from ensemble model integrations", *Journal of Climate*, vol. 9, no. 7, pp. 1518–1530, 1996. DOI: 10.1175/1520-0442(1996)009<1518:AMFPAE>2.0.CO;2.

[77] A. Jordan, F. Krüger, and S. Lerch, "Evaluating probabilistic forecasts with scoringrules", 2017. arXiv: 1709.04743.

[78] J. E. Matheson and R. L. Winkler, "Scoring rules for continuous probability distributions", *Management Science*, vol. 22, no. 10, pp. 1087–1096, 1976. DOI: 10.1287/mnsc.22.10.1087.

[79] R. L. Winkler, "A decision-theoretic approach to interval estimation", *Journal of the American Statistical Association*, vol. 67, pp. 187–191, 1972. DOI: 10.1080/01621459.1972.10481224.

[80] C. Kamath, "Associating weather conditions with ramp events in wind power generation", in *2011 IEEE/PES Power Systems Conference and Exposition*, IEEE, 2011, pp. 1–8. DOI: 10.1109/PSCE.2011.5772527.

[81] W.-B. Yu, H. Min, and B.-R. Lea, "A multiple-agent based system for forecasting the ice cream demand using climatic information", in *Management Intelligent Systems: First International Symposium*, Berlin, Germany: Springer, 2012, pp. 227–238.

[82] M. D. Agnew and J. E. Thornes, "The weather sensitivity of the UK food retail and distribution industry", *Meteorological Applications*, vol. 2, no. 2, pp. 137–147, 1995. DOI: 10.1002/met.5060020207.

[83] J. Oh, K.-J. Ha, and Y.-H. Jo, "New normal weather breaks a traditional clothing retail calendar", *Research Square*, 2021. DOI: 10.21203/rs.3.rs-213444/v1.

[84] E. N. Lorenz, "Deterministic nonperiodic flow", *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[85] A. C. Lorenc, "Analysis methods for numerical weather prediction", *Quarterly Journal of the Royal Meteorological Society*, vol. 112, no. 474, pp. 1177–1194, 1986. DOI: 10.1002/qj.49711247414.

[86] B.-W. Shen, R. A. Pielke Sr, X. Zeng, J.-J. Baik, S. Faghih-Naini, J. Cui, R. Atlas, and T. Reyes, "Is weather chaotic? Coexisting chaotic and non-chaotic attractors within lorenz models", in *13th Chaotic Modeling and Simulation International Conference 13*, Springer, 2021, pp. 805–825. DOI: 10.1007/978-3-030-70795-8_57.

[87] J. Coiffier, *Fundamentals of numerical weather prediction*. Cambridge, UK: Cambridge University Press, 2011.

[88] F. Molteni, R. Buizza, T. N. Palmer, and T. Petroliagis, "The ECMWF ensemble prediction system: Methodology and validation", *Quarterly Journal of the Royal Meteorological Society*, vol. 122, no. 529, pp. 73–119, 1996. DOI: 10.1002/qj.49712252905.

[89] N. E. Bowler, A. Arribas, K. R. Mylne, K. B. Robertson, and S. E. Beare, "The MOGREPS short-range ensemble prediction system", *Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric Sciences, Applied Meteorology and Physical Oceanography*, vol. 134, no. 632, pp. 703–722, 2008. DOI: 10.1002/qj.234.

[90] T. L. Thorarinsdottir and T. Gneiting, "Probabilistic forecasts of wind speed: Ensemble model output statistics by using heteroscedastic censored regression", *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 173, no. 2, pp. 371–388, 2010. DOI: 10.1111/j.1467-985x.2009.00616.x.

[91] R. Schefzik, T. L. Thorarinsdottir, and T. Gneiting, "Uncertainty quantification in complex simulation models using ensemble copula coupling", *Statistical Science*, vol. 28, no. 4, pp. 616–640, 2013. DOI: 10.1214/13-STS443.

[92] R. Schefzik, "Ensemble calibration with preserved correlations: Unifying and comparing ensemble copula coupling and member-by-member postprocessing", *Quarterly Journal of the Royal Meteorological Society*, vol. 143, no. 703, pp. 999–1008, 2017. DOI: 10.1002/qj.2984.

[93] T. Gneiting, "Calibration of medium-range weather forecasts", Technical Memorandum 719. ECMWF, Tech. Rep., 2014. DOI: 10.21957/8xna7glta.

[94] T. Gneiting, A. E. Raftery, A. H. Westveld, and T. Goldman, "Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation", *Monthly Weather Review*, vol. 133, no. 5, pp. 1098–1118, 2005. DOI: 10.1175/mwr2904.1.

[95] C. Gilbert, J. W. Messner, P. Pinson, P.-.-J. Trombe, R. Verzijlbergh, P. Dorp, and H. Jonker, "Statistical post–processing of turbulence–resolving weather forecasts for offshore wind power forecasting", *Wind Energy*, vol. 23, no. 4, pp. 884–897, 2020. DOI: 10.1002/we.2456.

[96] C. Sweeney and P. Lynch, "Adaptive post-processing of short-term wind forecasts for energy applications", *Wind Energy*, vol. 14, no. 3, pp. 317–325, 2011. DOI: 10.1002/we.420.

[97] A. Bossavy, R. Girard, and G. Kariniotakis, "Forecasting ramps of wind power production with numerical weather prediction ensembles", *Wind Energy*, vol. 16, no. 1, pp. 51–63, 2013. DOI: 10.1002/we.526.

[98] B. Schulz, M. El Ayari, S. Lerch, and S. Baran, "Post-processing numerical weather prediction ensembles for probabilistic solar irradiance forecasting", *Solar Energy*, vol. 220, pp. 1016–1031, 2021.

[99] S. Vannitsem, J. B. Bremnes, J. Demaeyer, G. R. Evans, J. Flowerdew, S. Hemri, S. Lerch, N. Roberts, S. Theis, A. Atencia, Z. B. Bouallègue, J. Bhend, M. Dabernig, L. D. Cruz, L. Hieta, O. Mestre, L. Moret, I. O. Plenkovi, M. Schmeits, M. Taillardat, J. V. den Bergh, B. V. Schaeybroeck, K. Whan, and J. Ylhaisi, "Statistical postprocessing for weather forecasts: Review, challenges, and avenues in a big data world", *Bulletin of the American Meteorological Society*, vol. 102, no. 3, E681–E699, 2021. DOI: 10.1175/bams-d-19-0308.1.

[100] P. Pinson and J. W. Messner, "Chapter 9 - application of postprocessing for renewable energy", in *Statistical Postprocessing of Ensemble Forecasts*. S. Vannitsem, D. S. Wilks, and J. W. Messner, Eds., Amsterdam, Netherlands: Elsevier, 2018, pp. 241–266.

[101] J. W. Taylor, P. E. McSharry, and R. Buizza, "Wind power density forecasting using ensemble predictions and time series models", *IEEE Transactions on Energy Conversion*, vol. 24, no. 3, pp. 775–782, 2009. DOI: 10.1109/tec.2009.2025431.

[102] P. Pinson and H. Madsen, "Ensemble-based probabilistic forecasting at horns rev", *Wind Energy*, vol. 12, no. 2, pp. 137–155, 2009. DOI: 10.1002/we.309.

[103] J. W. Messner, A. Zeileis, J. Broecker, and G. J. 5r, "Probabilistic wind power forecasts with an inverse power curve transformation and censored regression", *Wind Energy*, vol. 17, no. 11, pp. 1753–1766, 2014. DOI: 10.1002/we.1666.

[104] R. P. Worsnop, M. Scheuerer, T. M. Hamill, and J. K. Lundquist, "Generating wind power scenarios for probabilistic ramp event prediction using multivariate statistical post-processing", *Wind Energy Science*, vol. 3, no. 1, pp. 371–393, 2018. DOI: 10.5194/wes-3-371-2018.

[105] R. Williams, C. Ferro, and F. Kwasniok, "A comparison of ensemble post-processing methods for extreme events", *Quarterly Journal of the Royal Meteorological Society*, vol. 140, no. 680, pp. 1112–1120, 2014. DOI: 10.1002/qj.2198.

[106] S. Lerch, S. Baran, A. Möller, J. Gross, R. Schefzik, S. Hemri, and M. Graeter, "Simulation-based comparison of multivariate ensemble post-processing methods", *Nonlinear Processes in Geophysics*, vol. 27, no. 2, pp. 349–371, 2020. DOI: 10.5194/npg-27-349-2020.

[107] M. Taillardat and O. Mestre, "From research to applications–examples of operational ensemble post-processing in France using machine learning", *Nonlinear Processes in Geophysics*, vol. 27, no. 2, pp. 329–347, 2020. DOI: 10.5194/npg-27-329-2020.

[108] H. Poincaré, *Science and method*. Mineola, NY, USA: Dover Publications, 2003.

[109] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using Bayesian model averaging to calibrate forecast ensembles", *Monthly Weather Review*, vol. 133, no. 5, pp. 1155–1174, 2005. DOI: 10.1175/mwr2906.1.

[110] Z. Javanshiri, M. Fathi, and A. Mohammadi, "Comparison of the BMA and EMOS statistical methods for probabilistic quantitative precipitation forecasting", *Meteorological Applications*, vol. 28, 2021. DOI: 10.1002/met.1974.

[111] K. Han, J. Choi, and C. Kim, "Comparison of statistical post-processing methods for probabilistic wind speed forecasting", *Asia-Pacific Journal of Atmospheric Sciences*, vol. 54, no. 1, pp. 91–101, 2018. DOI: 10.1007/s13143-017-0062-z.

[112] S. Baran, A. Horányi, and D. Nemoda, "Comparison of BMA and EMOS statistical calibration methods for temperature and wind speed ensemble weather prediction", *Quarterly Journal of the Hungarian Meteorological Service*, no. 3, pp. 217–241, 2014.

[113] S. Vannitsem, D. S. Wilks, and J. Messner, *Statistical postprocessing of ensemble forecasts*. Amsterdam, Netherlands: Elsevier, 2018.

[114] C. Fraley, A. E. Raftery, and T. Gneiting, "Calibrating multimodel forecast ensembles with exchangeable and missing members using Bayesian model averaging", *Monthly Weather Review*, vol. 138, no. 1, pp. 190–202, 2010. DOI: 10.1175/2009mwr3046.1.

[115] J. Bröcker and H. Kantz, "The concept of exchangeability in ensemble forecasting", *Nonlinear Processes in Geophysics*, vol. 18, no. 1, pp. 1–5, 2011. DOI: 10.5194/npg-18-1-2011.

[116] I. Staffell and S. Pfenninger, "Using bias-corrected reanalysis to simulate current and future wind power output", *Energy*, vol. 114, pp. 1224–1239, 2016. DOI: 10.1016/j.energy.2016.08.068.

[117] R. Swinbank, M. Kyouda, P. Buchanan, L. Froude, T. M. Hamill, T. D. Hewson, J. H. Keller, M. Matsueda, J. Methven, F. Pappenberger, M. Scheuerer, H. A. Titley, L. Wilson, and M. Yamaguchi, "The TIGGE project and its achievements", *Bulletin of the American Meteorological Society*, vol. 97, no. 1, pp. 49–67, 2016. DOI: 10.1175/bams-d-13-00191.1.

[118] Copernicus Climate Change Service (C3S), *ERA5: Fifth generation of ECMWF atmospheric reanalyses of the global climate*, Copernicus Climate Change Service Climate Data Store (CDS), Ed., Copernicus Climate Change Service, 2017.

[119] J. Olauson and M. Bergkvist, "Modelling the Swedish wind power production using MERRA reanalysis data", *Renewable Energy*, vol. 76, pp. 717–725, 2015. DOI: 10.1016/j.renene.2014.11.085.

[120] Energy Market Inspectorate, "Elområden i sverige – analys av utvecklingen och konsekvenserna på marknaden, EIR 2012:06", EI R2012:06, 2012.

[121] Elområden.se. "Din guide till nya elmarknaden". Accessed on 2019-12-16. (2011), [Online]. Available: https://elomraden.se/om.

[122] European Network of Transmission System Operators (ENTSO-E). "Transparency platform (TP)". Accessed on 2023-10-02. (2019), [Online]. Available: https://transparency.entsoe.eu/.

[123] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression", in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, PMLR, 2018, pp. 2796–2804. DOI: 10.48550/arXiv.1807.00263.

[124] Y.-K. Wu and J.-S. Hong, "A literature review of wind forecasting technology in the world", in *2007 IEEE Lausanne POWERTECH, Proceedings*, IEEE, 2007, pp. 504–509. DOI: 10.1109/PCT.2007.4538368.

[125] M. Lydia and S. S. Kumar, "A comprehensive overview on wind power forecasting", in *2010 Conference Proceedings IPEC*, IEEE, 2010. DOI: 10.1109/ipecon.2010.5697118.

[126] Y. Zhang and J. Wang, "K-nearest neighbors and a kernel density estimator for GEFCom2014 probabilistic wind power forecasting", *International Journal of Forecasting*, vol. 32, no. 3, pp. 1074–1080, 2016. DOI: 10.1016/j.ijforecast.2015.11.006.

[127] J. E. Angus, "The probability integral transform and related results", *SIAM Review*, vol. 36, no. 4, pp. 652–654, 1994. DOI: 10.1137/1036146.

[128] T. M. Hamill, "Interpretation of rank histograms for verifying ensemble forecasts", *Monthly Weather Review*, vol. 129, no. 3, pp. 550–560, 2001. DOI: 10.1175/1520-0493(2001)129<0550:iorhfv>2.0.co;2.

[129] O. Grothe, F. Kächele, and F. Krüger, "From point forecasts to multivariate probabilistic forecasts: The schaake shuffle for day-ahead electricity price forecasting", *Energy Economics*, vol. 120, p. 106 602, 2023.

[130] A. E. Raftery, "Use and communication of probabilistic forecasts", *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 9, no. 6, pp. 397–410, 2016. DOI: 10.1002/sam.11302.

[131] L. Dannecker, *Energy time series forecasting: Efficient and accurate forecasting of evolving time series from the energy domain*, 1st ed. Wiesbaden, Germany: Springer Vieweg, 2015.

[132] J. Liu, N. Wu, Y. Qiao, and Z. Li, "A scientometric review of research on traffic forecasting in transportation", *IET Intelligent Transport Systems*, vol. 15, no. 1, pp. 1–16, 2021. DOI: 10.1049/itr2.12024.

[133] A. Koochali, P. Schichtel, A. Dengel, and S. Ahmed, "Probabilistic forecasting of sensory data with generative adversarial networks – ForGAN", *IEEE Access*, vol. 7, pp. 63 868–63 880, 2019. DOI: 10.1109/ACCESS.2019.2915544.

[134] F. Petropoulos, D. Apiletti, V. Assimakopoulos, M. Z. Babai, D. K. Barrow, S. B. Taieb, C. Bergmeir, R. J. Bessa, J. Bijak, J. E. Boylan, *et al.*, "Forecasting: Theory and practice", *International Journal of Forecasting*, vol. 38, no. 3, pp. 705–871, 2022. DOI: 10.1016/j.ijforecast.2021.11.001.

[135] W. H. Williams and M. Goodman, "A simple method for the construction of empirical confidence limits for economic forecasts", *Journal of the American Statistical Association*, vol. 66, pp. 752–754, 1971. DOI: 10.2307/2284223.

[136] E. Camporeale, X. Chu, O. Agapitov, and J. Bortnik, "On the generation of probabilistic forecasts from deterministic models", *Space Weather*, vol. 17, no. 3, pp. 455–475, 2019. DOI: 10.1029/2018SW002026.

[137] Y. Wang, G. Hug, Z. Liu, and N. Zhang, "Modeling load forecast uncertainty using generative adversarial networks", *Electric Power Systems Research*, vol. 189, p. 106 732, 2020. DOI: 10.1016/j.epsr.2020.106732.

[138] R. Krzysztofowicz, "Bayesian theory of probabilistic forecasting via deterministic hydrologic model", *Water Resources Research*, vol. 35, no. 9, pp. 2739–2750, 1999. DOI: 10.1029/1999WR900099.

[139] R. E. Caflisch, "Monte Carlo and quasi-Monte Carlo methods", *Acta Numerica*, vol. 7, pp. 1–49, 1998. DOI: 10.1017/S0962492900002804.

[140] R. C. Smith, *Uncertainty quantification: theory, implementation, and applications*. Philadelphia, PA, USA: SIAM, 2013, vol. 12.

[141] K. Stankeviciute, A. M Alaa, and M. van der Schaar, "Conformal time-series forecasting", in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 6216–6228.

[142] V. Chernozhukov, K. Wüthrich, and Y. Zhu, "Distributional conformal prediction", in *Proceedings of the National Academy of Sciences*, vol. 118, National Acad Sciences, 2021, e2107794118. DOI: 10.48550/arXiv.1909.07889.

[143] M. Zaffran, O. Féron, Y. Goude, J. Josse, and A. Dieuleveut, "Adaptive conformal predictions for time series", in *International Conference on Machine Learning*, PMLR, 2022, pp. 25 834–25 866. DOI: 10.48550/arXiv.2202.07282.

[144] D. Kaplan and M. Huang, "Bayesian probabilistic forecasting with large-scale educational trend data: A case study using NAEP", *Large-scale Assessments in Education*, vol. 9, no. 1, pp. 1–31, 2021. DOI: 10.1186/s40536-021-00108-2.

[145] A. E. Raftery, D. Madigan, and J. A. Hoeting, "Bayesian model averaging for linear regression models", *Journal of the American Statistical Association*, vol. 92, no. 437, pp. 179–191, 1997. DOI: 10.1080/01621459.1997.10473615.

[146] E. Y. Cramer, E. L. Ray, V. K. Lopez, J. Bracher, A. Brennen, A. J. C. Rivadeneira, A. Gerding, T. Gneiting, K. H. House, Y. Huang, *et al.*, "Evaluation of individual and ensemble probabilistic forecasts of COVID-19 mortality in the US", in *Proceedings of the National Academy of Sciences*, vol. 119, National Academy of Sciences, 2022, e2113561119. DOI: 10.1073/pnas.2113561119.

[147] E. Camporeale, A. Agnihotri, and C. Rutjes, "Adaptive selection of sampling points for uncertainty quantification", *International Journal for Uncertainty Quantification*, vol. 7, no. 4, pp. 1–22, 2017. DOI: 10.48550/arXiv.1612.07827.

[148] D. Xiu, *Numerical methods for stochastic computations*. Princeton, NJ, USA: Princeton University Press, 2010.

[149] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, and U. Köthe, "Guided image generation with conditional invertible neural networks", 2019. arXiv: 1907.02392.

[150] M. Arpogaus, M. Voss, B. Sick, M. Nigge-Uricher, and O. Dürr, "Short-term density forecasting of low-voltage load using Bernstein-polynomial normalizing flows", 2022. arXiv: 2204.13939.

[151] K. Rasul, A.-S. Sheikh, I. Schuster, U. Bergmann, and R. Vollgraf, "Multivariate probabilistic time series forecasting via conditioned normalizing flows", 2020. arXiv: 2002.06103.

[152] R. Wen and K. Torkkola, "Deep generative quantile-copula models for probabilistic forecasting", 2019. arXiv: 1907.10697.

[153] A. Jamgochian, D. Wu, K. Menda, S. Jung, and M. J. Kochenderfer, "Conditional approximate normalizing flows for joint multi-step probabilistic electricity demand forecasting", 2022. arXiv: 2201.02753.

[154] E. Cramer, D. Witthaut, A. Mitsos, and M. Dahmen, "Multivariate probabilistic forecasting of intraday electricity prices using normalizing flows", *Applied Energy*, vol. 346, p. 121 370, 2023. DOI: 10.1016/j.apenergy.2023.121370.

[155] J. Dumas, A. Wehenkel, D. Lanaspeze, B. Cornélusse, and A. Sutera, "A deep generative model for probabilistic energy forecasting in power systems: Normalizing flows", *Applied Energy*, vol. 305, p. 117 871, 2022. DOI: 10.1016/j.apenergy.2021.117871.

[156] L. Zhang and B. Zhang, "Scenario forecasting of residential load profiles", *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 1, pp. 84–95, 2019. DOI: 10.1109/JSAC.2019.2951973.

[157] L. Ge, W. Liao, S. Wang, B. Bak-Jensen, and J. R. Pillai, "Modeling daily load profiles of distribution network for scenario generation using flow-based generative network", *IEEE Access*, vol. 8, pp. 77 587–77 597, 2020. DOI: 10.1109/ACCESS.2020.2989350.

[158] A. Fanfarillo, B. Roozitalab, W. Hu, and G. Cervone, "Probabilistic forecasting using deep generative models", *GeoInformatica*, vol. 25, no. 1, pp. 127–147, 2021. DOI: 10.1007/s10707-020-00425-8.

[159] C. De La Vallée Poussin, "Sur lintégrale de lebesgue", *Transactions of the American Mathematical Society*, vol. 16, no. 4, pp. 435–501, 1915.

[160] K. P. Murphy, *Probabilistic Machine Learning: Advanced Topics*. Cambridge, MA, USA: MIT Press, 2023.

[161] D. Dua and C. Graff. "UCI machine learning repository". Accessed on 2022-03-10. (2019), [Online]. Available: http://archive.ics.uci.edu/ml.

[162] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond", *International Journal of Forecasting*, vol. 32, no. 3, pp. 896–913, 2016. DOI: 10.1016/j.ijforecast.2016.02.001.

[163] H. Fanaee-T and J. Gama, "Event labeling combining ensemble detectors and background knowledge", *Progress in Artificial Intelligence*, vol. 2, pp. 113–127, 2014. DOI: 10.1007/s13748-013-0040-3.

[164] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza, M. Mergenthaler, and A. Dubrawski, "N-HITS: Neural hierarchical interpolation for time series forecasting", 2022. arXiv: 2201.12886.

[165] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. DOI: 10.48550/arXiv.1201.0490.

[166] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *3rd International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., IEEE, 2015. DOI: 10.48550/arXiv.1412.6980.

[167] Martín Abadi *et al.* "TensorFlow: Large-scale machine learning on heterogeneous systems". Accessed on 2023-10-02. (2015), [Online]. Available: https://www.tensorflow.org/.

[168] F. Chollet. "Keras". Accessed on 2023-10-02. (2015), [Online]. Available: https://github.com/fchollet/keras.

[169] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system", in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[170] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks", *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020. DOI: 10.1016/j.ijforecast.2019.07.001.

[171] R. Koenker, V. Chernozhukov, X. He, and L. Peng, *Handbook of Quantile Regression*. Boca Raton, FL, USA: CRC press, 2017.

[172] González Ordiano, J. A., L. Gröll, R. Mikut, and V. Hagenmeyer, "Probabilistic energy forecasting using the nearest neighbors quantile filter and quantile regression", *International Journal of Forecasting*, vol. 36, no. 2, pp. 310–323, 2020. DOI: 10.1016/j.ijforecast.2019.06.003.

[173] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions", in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018, pp. 10 215–10 224. DOI: /10.48550/arXiv.1807.03039.

[174] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation", 2014. arXiv: 1410.8516.

[175] O. J. Dunn, "Multiple comparisons among means", *Journal of the American statistical association*, vol. 56, no. 293, pp. 52–64, 1961.

[176] C. Sweeney, R. J. Bessa, J. Browell, and P. Pinson, "The future of forecasting for renewable energy", *WIREs Energy and Environment*, vol. 9, no. 2, e365, 2020. DOI: https://doi.org/10.1002/wene.365.

[177] T. Duan, A. Anand, D. Y. Ding, K. K. Thai, S. Basu, A. Ng, and A. Schuler, "Ngboost: Natural gradient boosting for probabilistic prediction", in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, 2020, pp. 2690–2700. DOI: 10.48550/arXiv.1910.03225.

[178] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network", *Neurocomputing*, vol. 399, pp. 491–501, 2020. DOI: 10.1016/j.neucom.2020.03.011.

[179] J. Gasthaus, K. Benidis, Y. Wang, S. S. Rangapuram, D. Salinas, V. Flunkert, and T. Januschowski, "Probabilistic forecasting with spline quantile function RNNs", in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, K. Chaudhuri and M. Sugiyama, Eds., PMLR, 2019, pp. 1901–1910.

[180] L. Xu, S. Wang, and R. Tang, "Probabilistic load forecasting for buildings considering weather forecasting uncertainty and uncertain peak load", *Applied Energy*, vol. 237, pp. 180–195, 2019. DOI: 10.1016/j.apenergy.2019.01.022.

[181] E. Choi, S. Cho, and D. K. Kim, "Power demand forecasting using long short-term memory (lstm) deep-learning model for monitoring energy sustainability", *Sustainability*, vol. 12, no. 3, p. 1109, 2020. DOI: 10.3390/su12031109.

[182] Y. Yang, W. Li, T. A. Gulliver, and S. Li, "Bayesian deep learning-based probabilistic load forecasting in smart grids", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4703–4713, 2020. DOI: 10.1109/TII.2019.2942353.

[183] M. Kostrzewski and J. Kostrzewska, "Probabilistic electricity price forecasting with Bayesian stochastic volatility models", *Energy Economics*, vol. 80, pp. 610–620, 2019. DOI: 10.1016/j.eneco.2019.02.004.

[184] G. Marcjasz, B. Uniejewski, and R. Weron, "Probabilistic electricity price forecasting with NARX networks: Combine point or probabilistic forecasts?", *International Journal of Forecasting*, vol. 36, no. 2, pp. 466–479, 2020. DOI: 10.1016/j.ijforecast.2019.07.002.

[185] H. Zhang, Y. Liu, J. Yan, S. Han, L. Li, and Q. Long, "Improved deep mixture density network for regional wind power probabilistic forecasting", *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2549–2560, 2020. DOI: 10.1109/TPWRS.2020.2971607.

[186] K. Wang, Y. Zhang, F. Lin, J. Wang, and M. Zhu, "Nonparametric probabilistic forecasting for wind power generation using quadratic spline quantile function and autoregressive recurrent neural network", *IEEE Transactions on Sustainable Energy*, vol. 13, no. 4, pp. 1930–1943, 2022. DOI: 10.1109/TSTE.2022.3175916.

[187] M. Sun, C. Feng, and J. Zhang, "Probabilistic solar power forecasting based on weather scenario generation", *Applied Energy*, vol. 266, p. 114 823, 2020. DOI: 10.1016/j.apenergy.2020.114823.

[188] H. Panamtash, Q. Zhou, T. Hong, Z. Qu, and K. O. Davis, "A copula-based Bayesian method for probabilistic solar power forecasting", *Solar Energy*, vol. 196, pp. 336–345, 2020. DOI: 10.1016/j.solener.2019.11.079.

[189] J. Huber, D. Dann, and C. Weinhardt, "Probabilistic forecasts of time and energy flexibility in battery electric vehicle charging", *Applied Energy*, vol. 262, p. 114 525, 2020. DOI: 10.1016/j.apenergy.2020.114525.

[190] S. Baran and S. Lerch, "Mixture EMOS model for calibrating ensemble forecasts of wind speed", *Environmetrics*, vol. 27, pp. 116–130, 2016. DOI: 10.1002/env.2380.

[191] P. Grönquist, C. Yao, T. Ben-Nun, N. Dryden, P. Dueben, S. Li, and T. Hoefler, "Deep learning for post-processing ensemble weather forecasts", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2194, p. 20 200 092, 2021. DOI: 10.1098/rsta.2020.0092.

[192] A. Alsharef, K. Aggarwal, M. Kumar, and A. Mishra, "Review of ML and AutoML solutions to forecast time-series data", *Archives of Computational Methods in Engineering*, vol. 29, no. 7, pp. 5297–5311, 2022. DOI: 10.1007/s11831-022-09765-0.

[193] S. S. Rangapuram, L. D. Werner, K. Benidis, P. Mercado, J. Gasthaus, and T. Januschowski, "End-to-end learning of coherent probabilistic forecasts for hierarchical time series", in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., vol. 139, PMLR, 2021, pp. 8832–8843.

[194] O. Shchur, C. Turkmen, N. Erickson, H. Shen, A. Shirkov, T. Hu, and Y. Wang, "Autogluon-timeseries: AutoML for probabilistic time series forecasting", 2023. arXiv: 2308.05566.

[195] W. Zhao, H. Zhang, J. Zheng, Y. Dai, L. Huang, W. Shang, and Y. Liang, "A point prediction method based automatic machine learning for day-ahead power output of multi-region photovoltaic plants", *Energy*, vol. 223, p. 120 026, 2021. DOI: 10.1016/j.energy.2021.120026.

[196] S. Meisenbacher, B. Heidrich, T. Martin, R. Mikut, and V. Hagenmeyer, "AutoPV: Automated photovoltaic forecasts with limited information using an ensemble of pre-trained models", 2022. arXiv: 2212.06797.

[197] S. M. J. Jalali, S. Ahmadian, A. Kavousi-Fard, A. Khosravi, and S. Nahavandi, "Automated deep CNN-LSTM architecture design for solar irradiance forecasting", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 54–65, 2022. DOI: 10.1109/TSMC.2021.3093519.

[198] C. N. Bergmeir, "New approaches in time series forecasting: Methods, software, and evaluation procedures", Ph.D. dissertation, Universidad de Granada, Granada, Spain, 2013.

[199] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training", 2018. arXiv: 1807.05118.

[200] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization", in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., vol. 24, Curran Associates, Inc., 2011.

[201] T. Januschowski, J. Gasthaus, Y. Wang, D. Salinas, V. Flunkert, M. Bohlke-Schneider, and L. Callot, "Criteria for classifying forecasting methods", *International Journal of Forecasting*, vol. 36, no. 1, pp. 167–177, 2020. DOI: 10.1016/j.ijforecast.2019.05.008.

[202] Y. Liang, S. Li, C. Yan, M. Li, and C. Jiang, "Explaining the black-box model: A survey of local interpretation methods for deep neural networks", *Neurocomputing*, vol. 419, pp. 168–182, 2021. DOI: 10.1016/j.neucom.2020.08.011.

[203] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models", *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018. DOI: 10.1145/3236009.

[204] B. Tang and D. S. Matteson, "Probabilistic transformer for time series analysis", vol. 34, Curran Associates, Inc., 2021, pp. 23 592–23 608.

[205] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, Association for the Advancement of Artificial Intelligence, 2021, pp. 11 106–11 115. DOI: 10.48550/arXiv.2012.07436.

[206] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "XAI – explainable artificial intelligence", *Science Robotics*, vol. 4, no. 37, eaay7120, 2019. DOI: 10.1126/scirobotics.aay7120.

[207] A. B. Arrieta, N. Daz-Rodrguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garca, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai", *Information Fusion*, vol. 58, pp. 82–115, 2020. DOI: 10.1016/j.inffus.2019.12.012.

[208] F. K. Doilovi, M. Bri, and N. Hlupi, "Explainable artificial intelligence: A survey", in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2018, pp. 0210–0215. DOI: 10.23919/MIPRO.2018.8400040.

[209] J. Castro, D. Gómez, and J. Tejada, "Polynomial calculation of the Shapley value based on sampling", *Computers & Operations Research*, vol. 36, no. 5, pp. 1726–1730, 2009. DOI: 10.1016/j.cor.2008.04.004.

[210] C. Molnar, *Interpretable machine learning*. Durham, NC, USA: Lulu Press, Inc., 2020.

[211] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks", in *International Conference on Machine Learning*, PMLR, 2017, pp. 3319–3328. DOI: 10.48550/arXiv.1703.01365.

[212] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps", 2013. arXiv: 1312.6034.

[213] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting", 2019. arXiv: 1905.10437.

[214] J. Wang, Z. Wang, J. Li, and J. Wu, "Multilevel wavelet decomposition network for interpretable time series analysis", in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, 2018, pp. 2437–2446. DOI: 10.1145/3219819.3220060.

[215] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, "Towards a rigorous evaluation of XAI methods on time series", in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, IEEE, 2019, pp. 4197–4201. DOI: 10.1109/ICCVW.2019.00516.

[216] A. Theissler, F. Spinnato, U. Schlegel, and R. Guidotti, "Explainable AI for time series classification: A review, taxonomy and research directions", *IEEE Access*, 2022. DOI: 10.1109/ACCESS.2022.3207765.

[217] R. Mochaourab, A. Venkitaraman, I. Samsten, P. Papapetrou, and C. R. Rojas, "Post hoc explainability for time series classification: Toward a signal processing perspective", *IEEE Signal Processing Magazine*, vol. 39, no. 4, pp. 119–129, 2022. DOI: 10.1109/MSP.2022.3155955.

[218] E. M. Kenny, E. D. Delaney, D. Greene, and M. T. Keane, "Post-hoc explanation options for XAI in deep learning: The insight centre for data analytics perspective", in *Pattern Recognition. ICPR International Workshops and Challenges, Proceedings, Part III*, Springer, 2021, pp. 20–34. DOI: 10.1007/978-3-030-68796-0_2.

[219] H. Turbé, M. Bjelogrlic, C. Lovis, and G. Mengaldo, "Evaluation of post-hoc interpretability methods in time-series classification", *Nature Machine Intelligence*, vol. 5, no. 3, pp. 250–260, 2023. DOI: 10.1038/s42256-023-00620-w.

[220] F. Di Martino and F. Delmastro, "Explainable AI for clinical and remote health applications: A survey on tabular and time series data", *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5261–5315, 2023. DOI: 10.1007/s10462-022-10304-3.

[221] S. Tonekaboni, S. Joshi, K. Campbell, D. K. Duvenaud, and A. Goldenberg, "What went wrong and when? Instance-wise feature importance for time-series black-box models", vol. 33, Curran Associates, Inc., 2020, pp. 799–809.

[222] M. Munir, S. A. Siddiqui, F. Küsters, D. Mercier, A. Dengel, and S. Ahmed, "TSXplain: Demystification of DNN decisions for time-series using natural language and statistical features", in *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks*, Springer, 2019, pp. 426–439. DOI: 10.1007/978-3-030-30493-5_43.

[223] Q. Pan, W. Hu, and N. Chen, "Two birds with one stone: Series saliency for accurate and interpretable multivariate time series forecasting.", in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, IJCAI, 2021, pp. 2884–2891. DOI: 10.24963/ijcai.2021/397.

[224] M. Hertel, S. Ott, B. Schäfer, R. Mikut, V. Hagenmeyer, and O. Neumann, "Evaluation of transformer architectures for electrical load time-series forecasting", in *Proceedings - 32. Workshop Computational Intelligence*, vol. 1, KIT Scientific Publishing, 2022, p. 93. DOI: 10.5445/IR/1000154155.

[225] J. Kruse, B. Schäfer, and D. Witthaut, "Revealing drivers and risks for power grid frequency stability with explainable AI", *Patterns*, vol. 2, no. 11, 2021. DOI: 10.1016/j.patter.2021.100365.

[226] J. Trebbien, L. R. Gorjão, A. Praktiknjo, B. Schäfer, and D. Witthaut, "Understanding electricity prices beyond the merit order principle using explainable AI", *Energy and AI*, vol. 13, p. 100 250, 2023. DOI: 10.1016/j.egyai.2023.100250.

[227] H. Choi, C. Jung, T. Kang, H. J. Kim, and I.-Y. Kwak, "Explainable time-series prediction using a residual network and gradient-based methods", *IEEE Access*, vol. 10, pp. 108 469–108 482, 2022. DOI: 10.1109/ACCESS.2022.3213926.

[228] T. B. Çelik, Ö. can, and E. Bulut, "Extending machine learning prediction capabilities by explainable AI in financial time series prediction", *Applied Soft Computing*, vol. 132, p. 109 876, 2023. DOI: 10.1016/j.asoc.2022.109876.

[229] K. Wang, L. Zhang, and X. Fu, "Time series prediction of tunnel boring machine (TBM) performance during excavation using causal explainable artificial intelligence (CX-AI)", *Automation in Construction*, vol. 147, p. 104 730, 2023. DOI: 10.1016/j.autcon.2022.104730.

[230] A. Barredo Arrieta, S. Gil-Lopez, I. Laña, M. N. Bilbao, and J. Del Ser, "On the post-hoc explainability of deep echo state networks for time series forecasting, image and video classification", *Neural Computing and Applications*, pp. 1–21, 2022. DOI: 10.1007/s00521-021-06359-y.

[231] U. Von Luxburg and B. Schölkopf, "Statistical learning theory: Models, concepts, and results", in *Handbook of the History of Logic*, vol. 10, Elsevier, 2011, pp. 651–706.

[232] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Gradient-based attribution methods", in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Springer, 2019, pp. 169–191.

[233] A. Holzinger, A. Saranti, C. Molnar, P. Biecek, and W. Samek, "Explainable AI methods - a brief overview", in *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, Springer, 2020, pp. 13–38. DOI: 10.1007/978-3-031-04083-2_2.

[234] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, *Captum: A unified and generic model interpretability library for PyTorch*, 2020. arXiv: 2009.07896 [cs.LG].

[235] E. Strumbelj and I. Kononenko, "An efficient explanation of individual classifications using game theory", *The Journal of Machine Learning Research*, vol. 11, pp. 1–18, 2010.

[236] O. Neumann, N. Ludwig, M. Turowski, B. Heidrich, V. Hagenmeyer, and R. Mikut, "Smart data representations: Impact on the accuracy of deep neural networks", in *Proceedings 30. Workshop Computational Intelligence*, vol. 25, KIT Scientific Publishing, 2021, p. 113. DOI: 10.5445/KSP/1000138532.

[237] C. Rudin and J. Radin, "Why are we using black box models in AI when we dont need to? A lesson from an explainable AI competition", *Harvard Data Science Review*, vol. 1, no. 2, pp. 1–9, 2019. DOI: 10.1162/99608f92.5a8a3a3d.

[238] D. Banister, "Cities, mobility and climate change", *Journal of Transport Geography*, vol. 19, no. 6, pp. 1538–1546, 2011. DOI: 10.1016/j.jtrangeo.2011.03.009.

[239] A. Bick, A. Blandin, and K. Mertens. "Work from home after the COVID-19 outbreak". Accessed on 2023-10-02. (2020), [Online]. Available: https://doi.org/10.24149/wp2017.

[240] M. H. Eiza, Y. Cao, and L. Xu, *Toward sustainable and economic smart mobility: Shaping the future of smart cities*. Singapore, Singapore: World Scientific Publishing, 2020.

[241] D. Alghamdi, K. Basulaiman, and J. Rajgopal, "Multi-stage deep probabilistic prediction for travel demand", *Applied Intelligence*, vol. 52, no. 10, pp. 11 214–11 231, 2022. DOI: 10.1007/s10489-021-03047-1.

[242] K. Zhao, D. Khryashchev, and H. Vo, "Predicting taxi and uber demand in cities: Approaching the limit of predictability", *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2723–2736, 2019. DOI: 10.1109/TKDE.2019.2955686.

[243] X. Qian, S. V. Ukkusuri, C. Yang, and F. Yan, "Forecasting short-term taxi demand using boosting-GCRF", in *6th ACM SIGKDD International Workshop on Urban Computing*, vol. 14, ACM, 2017, pp. 272–285.

[244] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks", *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2572–2581, 2017. DOI: 10.1109/TITS.2017.2755684.

[245] T. Afrin and N. Yodo, "A probabilistic estimation of traffic congestion using Bayesian network", *Measurement*, vol. 174, p. 109 051, 2021. DOI: 10.1016/j.measurement.2021.109051.

[246] X. Feng, M. Saito, and Y. Liu, "Improve urban passenger transport management by rationally forecasting traffic congestion probability", *International Journal of Production Research*, vol. 54, no. 12, pp. 3465–3474, 2016. DOI: 10.1080/00207543.2015.1062570.

[247] H. Lim, K. Chung, and S. Lee, "Probabilistic forecasting for demand of a bike-sharing service using a deep-learning approach", *Sustainability*, vol. 14, no. 23, p. 15 889, 2022. DOI: 10.3390/su142315889.

[248] N. Gast, G. Massonnet, D. Reijsbergen, and M. Tribastone, "Probabilistic forecasts of bike-sharing systems for journey planning", in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ACM, 2015, pp. 703–712. DOI: 10.1145/2806416.2806569.

[249] H. Yu, Z. Wu, D. Chen, and X. Ma, "Probabilistic prediction of bus headway using relevance vector machine regression", *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1772–1781, 2016. DOI: 10.1109/TITS.2016.2620483.

[250] T. Bapaume, E. Côme, M. Ameli, J. Roos, and L. Oukhellou, "Forecasting passenger flows and headway at train level for a public transport line: Focus on atypical situations", *Transportation Research Part C: Emerging Technologies*, vol. 153, p. 104 195, 2023. DOI: 10.1016/j.trc.2023.104195.

[251] F. Caicedo, C. Blazquez, and P. Miranda, "Prediction of parking space availability in real time", *Expert Systems with Applications*, vol. 39, no. 8, pp. 7281–7290, 2012. DOI: 10.1016/j.eswa.2012.01.091.

[252] A. R. Hole, "Forecasting the demand for an employee Park and Ride service using commuters' stated choices", *Transport Policy*, vol. 11, no. 4, pp. 355–362, 2004. DOI: 10.1016/j.tranpol.2004.04.003.

[253] V. Miz and V. Hahanov, "Smart traffic light in terms of the cognitive road traffic management system (CTMS) based on the Internet of Things", in *Proceedings of IEEE East-West Design & Test Symposium (EWDTS 2014)*, IEEE, 2014, pp. 1–5. DOI: 10.1109/EWDTS.2014.7027102.

[254] O. Frendo, N. Gaertner, and H. Stuckenschmidt, "Improving smart charging prioritization by predicting electric vehicle departure time", *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6646–6653, 2020. DOI: 10.1109/TITS.2020.2988648.

[255] R. Cornell, "The climate change mitigation potential of electric vehicles as a function of renewable energy", *The International Journal of Climate Change: Impacts and Responses*, vol. 11, no. 1, pp. 15–24, 2019. DOI: 10.18848/1835-7156/cgp/v11i01/15-24.

[256] J. García-Villalobos, I. Zamora, J. San Martín, F. Asensio, and V. Aperribay, "Plug-in electric vehicles in electric distribution networks: A review of smart charging approaches", *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 717–731, 2014. DOI: https://doi.org/10.1016/j.rser.2014.07.040.

[257] A. S. Al-Ogaili *et al.*, "Review on scheduling, clustering, and forecasting strategies for controlling electric vehicle charging: Challenges and recommendations", *IEEE Access*, vol. 7, pp. 128 353–128 371, 2019. DOI: 10.1109/access.2019.2939595.

[258] Q. Wang, X. Liu, J. Du, and F. Kong, "Smart charging for electric vehicles: A survey from the algorithmic perspective", *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1500–1517, 2016. DOI: 10.1109/COMST.2016.2518628.

[259] K. Schwenk, "A smart charging assistant for electric vehicles considering battery degradation, power grid and user constraints", Ph.D. dissertation, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2022.

[260] K. Schwenk, T. Harr, R. Grossmann, R. R. Appino, V. Hagenmeyer, and R. Mikut, "Data-driven charging strategies for grid-beneficial, customer-oriented and battery-preserving electric mobility", in *Proceedings 29th Workshop Computational Intelligence*, KIT Scientific Publishing, 2019, pp. 73–93. DOI: 10.5445/IR/1000100252.

[261] M. Schmidt, P. Staudt, and C. Weinhardt, "Evaluating the importance and impact of user behavior on public destination charging of electric vehicles", *Applied Energy*, vol. 258, p. 114 061, 2020. DOI: 10.1016/j.apenergy.2019.114061.

[262] K. Schwenk, M. Faix, R. Mikut, V. Hagenmeyer, and R. R. Appino, "On calendar-based scheduling for user-friendly charging of plug-in electric vehicles", in *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*, IEEE, 2019, pp. 1–5. DOI: 10.1109/cavs.2019.8887782.

[263] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility", *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010. DOI: 10.1126/science.1177170.

[264] D. Austin, R. M. Cross, T. Hayes, and J. Kaye, "Regularity and predictability of human mobility in personal space", *PLoS One*, vol. 9, no. 2, e90256, 2014. DOI: 10.1371/journal.pone.0090256.

[265] Y. Cao, T. Wang, O. Kaiwartya, G. Min, N. Ahmad, and A. H. Abdullah, "An EV charging management system concerning drivers trip duration and mobility uncertainty", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 596–607, 2016. DOI: 10.1109/TSMC.2016.2613600.

[266] N. Daina, A. Sivakumar, and J. W. Polak, "Electric vehicle charging choices: Modelling and implications for smart charging services", *Transportation Research Part C: Emerging Technologies*, vol. 81, pp. 36–56, 2017. DOI: 10.1016/j.trc.2017.05.006.

[267] K. Schwenk, S. Meisenbacher, B. Briegel, T. Harr, V. Hagenmeyer, and R. Mikut, "Integrating battery aging in the optimization for bidirectional charging of electric vehicles", *IEEE Transaction on Smart Grid*, vol. 12, no. 6, pp. 5135–5145, 2021. DOI: 10.1109/TSG.2021.3099206.

[268] S. Meisenbacher, K. Schwenk, J. Galenzowski, S. Waczowicz, R. Mikut, and V. Hagenmeyer, "Smart charging of electric vehicles with cloud-based optimization and a lightweight User Interface: A real-world application in the Energy Lab 2.0: Poster", in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, ACM, 2021, pp. 284–285. DOI: 10.5445/IR/1000135104.

[269] C. Bikcora, L. Verheijen, and S. Weiland, "Density forecasting of daily electricity demand with ARMA-GARCH, CAViaR, and CARE econometric models", *Sustainable Energy, Grids and Networks*, vol. 13, no. 1, pp. 148–156, 2018. DOI: https://doi.org/10.1016/j.segan.2018.01.001.

[270] M. Amini, O. Karabasoglu, M. D. Ilic, K. G. Boroojeni, and S. S. Iyengar, "ARIMA-based demand forecasting method considering probabilistic model of electric vehicles' parking lots", in *2015 IEEE Power & Energy Society General Meeting*, IEEE, 2015, pp. 1–5. DOI: 10.1109/pesgm.2015.7286050.

[271] E. S. Xydas, C. E. Marmaras, L. M. Cipcigan, A. S. Hassan, and N. Jenkins, "Forecasting electric vehicle charging demand using support vector machines", in *2013 48th International Universities' Power Engineering Conference (UPEC)*, IEEE, 2013, pp. 1–6. DOI: 10.1109/upec.2013.6714942.

[272] M. B. Arias and S. Bae, "Electric vehicle charging demand forecasting model based on big data technologies", *Applied Energy*, vol. 183, pp. 327–339, 2016. DOI: 10.1016/j.apenergy.2016.08.080.

[273] D. Panahi, S. Deilami, and M. Masoum, "Forecasting plug-in electric vehicles load profile using artificial neural networks", IEEE, 2015, pp. 1–6. DOI: 10.1109/AUPEC.2015.7324879.

[274] B. Khaki, Y.-W. Chung, C. Chu, and R. Gadh, "Probabilistic electric vehicle load management in distribution grids", in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, IEEE, 2019, pp. 1–6. DOI: 10.1109/ITEC.2019.8790535.

[275] M. Amini and M. P. Moghaddam, "Probabilistic modelling of electric vehicles' parking lots charging demand", in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, IEEE, 2013, pp. 1–4. DOI: 10.1109/IranianCEE.2013.6599716.

[276] A. Ul-Haq, C. Cecati, and E. El-Saadany, "Probabilistic modeling of electric vehicle charging pattern in a residential distribution network", *Electric Power Systems Research*, vol. 157, pp. 126–133, 2018. DOI: 10.1016/j.epsr.2017.12.005.

[277] X. Zhang, K. W. Chan, H. Li, H. Wang, J. Qiu, and G. Wang, "Deep-learning-based probabilistic forecasting of electric vehicle charging load with a novel queuing model", *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3157–3170, 2020. DOI: 10.1109/TCYB.2020.2975134.

[278] A. T. Thorgeirsson, S. Scheubner, S. Fünfgeld, and F. Gauterin, "Probabilistic prediction of energy demand and driving range for electric vehicles with federated learning", *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 151–161, 2021. DOI: 10.1109/OJVT.2021.3065529.

[279] L. Buzna *et al.*, "An ensemble methodology for hierarchical probabilistic electric vehicle load forecasting at regular charging stations", *Applied Energy*, vol. 283, p. 116 337, 2021. DOI: 10.1016/j.apenergy.2020.116337.

[280] L. Petkevicius, S. Saltenis, A. Civilis, and K. Torp, "Probabilistic deep learning for electric-vehicle energy-use prediction", in *17th International Symposium on Spatial and Temporal Databases*, ACM, 2021, pp. 85–95. DOI: 10.1145/3469830.3470915.

[281] C. Goebel and M. Voss, "Forecasting driving behavior to enable efficient grid integration of plug-in electric vehicles", in *2012 IEEE Online Conference on Green Communications (GreenCom)*, IEEE, 2012, pp. 74–79. DOI: 10.1109/GreenCom.2012.6519619.

[282] H. Jahangir *et al.*, "Charging demand of plug-in electric vehicles: Forecasting travel behavior based on a novel rough artificial neural network approach", *Journal of Cleaner Production*, vol. 229, no. 20, pp. 1029–1044, 2019. DOI: https://doi.org/10.1016/j.jclepro.2019.04.345.

[283] O. Frendo, J. Graf, N. Gaertner, and H. Stuckenschmidt, "Data-driven smart charging for heterogeneous electric vehicle fleets", *Energy and AI*, vol. 1, p. 100 007, 2020. DOI: 10.1016/j.egyai.2020.100007.

[284] F. Zong, Y. Tian, Y. He, J. Tang, and J. Lv, "Trip destination prediction based on multi-day GPS data", *Physica A: Statistical Mechanics and its Applications*, vol. 515, pp. 258–269, 2019. DOI: 10.1016/j.physa.2018.09.090.

[285] Z. Zhao, H. N. Koutsopoulos, and J. Zhao, "Individual mobility prediction using transit smart card data", *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 19–34, 2018. DOI: 10.1016/j.trc.2018.01.022.

[286] A. Rossi, G. Barlacchi, M. Bianchini, and B. Lepri, "Modelling taxi drivers' behaviour for the next destination prediction", *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2980–2989, 2020. DOI: 10.1109/tits.2019.2922002.

[287] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[288] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, vol. 96, ACM, 1996.

[289] Q. Sun and Z. Ge, "Gated stacked target-related autoencoder: A novel deep feature extraction and layerwise ensemble method for industrial soft sensor application", *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3457–3468, 2020. DOI: 10.1109/TCYB.2020.3010331.

[290] Q. Zhu, "Latent variable regression for supervised modeling and monitoring", *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 800–811, 2020. DOI: 10.1109/JAS.2020.1003153.

[291] X. Shi, Q. Kang, J. An, and M. Zhou, "Novel L1 regularized extreme learning machine for soft-sensing of an industrial process", *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1009–1017, 2021. DOI: 10.1109/TII.2021.3065377.

[292] G. Wang and J. Qiao, "An efficient self-organizing deep fuzzy neural network for nonlinear system modeling", *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 7, pp. 2170–2182, 2021. DOI: 10.1109/TFUZZ.2021.3077396.

[293] K. Jiang, Z. Jiang, Y. Xie, D. Pan, and W. Gui, "Prediction of multiple molten iron quality indices in the blast furnace ironmaking process based on attention-wise deep transfer network", *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022. DOI: 10.1021/acsomega.2c05029.

[294] G. Wang, J. Bi, Q.-S. Jia, J. Qiao, and L. Wang, "Event-driven model predictive control with deep learning for wastewater treatment process", *IEEE Transactions on Industrial Informatics*, 2022. DOI: 10.1109/TII.2022.3177457.

[295] T. Minka, *Bayesian Linear Regression*. Pennsylvania, PA, USA: Citeseer, 2000.

[296] I. Castillo, J. Schmidt-Hieber, and A. Van der Vaart, "Bayesian linear regression with sparse priors", *The Annals of Statistics*, vol. 43, no. 5, pp. 1986–2018, 2015. DOI: 10.1214/15-AOS1334.

[297] Python Software Foundation. "Python". Accessed on 2023-10-02. (2019), [Online]. Available: https://www.python.org/.

[298] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine", *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[299] D. J. MacKay, "Bayesian interpolation", *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992. DOI: 10.1162/neco.1992.4.3.415.

[300] J. Q. Shi and T. Choi, *Gaussian Process Regression Analysis for Functional Data*. Boca Raton, FL, USA: CRC Press, 2011.

[301] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006, vol. 2.

[302] J. H. Friedman, "Greedy function approximation: A gradient boosting machine", *Annals of Statistics*, pp. 1189–1232, 2001.

[303] A. F. Agarap, "Deep learning using rectified linear units (ReLU)", 2018. arXiv: 1803.08375.

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

[305] R. Mikut, *Data Mining in der Medizin und Medizintechnik*. Karlsruhe, Germany: KIT Scientific Publishing, 2008, vol. 22.

[306] R. R. Appino, J. Á. G. Ordiano, R. Mikut, V. Hagenmeyer, and T. Faulwasser, "Storage scheduling with stochastic uncertainties: Feasibility and cost of imbalances", in *2018 Power Systems Computation Conference (PSCC)*, IEEE, 2018, pp. 1–7. DOI: 10.23919/PSCC.2018.8442529.

[307] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture", *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, 2020. DOI: 10.1109/JSEN.2020.3029459.

[308] F. Wiegel, J. Wachter, M. Kyesswa, R. Mikut, S. Waczowicz, and V. Hagenmeyer, "Smart energy system control laboratory–a fully-automated and user-oriented research infrastructure for controlling and operating smart energy systems", *at-Automatisierungstechnik*, vol. 70, no. 12, pp. 1116–1133, 2022. DOI: 10.1515/auto-2022-0018.

[309] S. Beichter, M. Beichter, D. Werling, G. Johannes, V. Weise, C. Hildenbrand, F. Wiegel, S. Waczowicz, R. Mikut, and V. Hagenmeyer, "Towards a real-world dispatchable feeder", in *Proceedings of the 8th IEEE Workshop on the Electronic Grid (eGrid 2023)*, IEEE, 2023, pp. 1–6.

A

Probabilistic Forecasts from Meteorological Uncertainty

To show the robust nature of our approach, we perform numerous further experiments. First, we implement both an alternative linear regression and a neural network as further forecasting methods, before also including the full results when considering different distributions for wind power and wind speed. Finally, we show that the selected probability distributions are a reasonable fit for our data and therefore a suitable choice for Ensemble Model Output Statistics (EMOS).

Alternative Linear Regression Model The linear regression proposed by Zhang and Wang [126] only considers wind speed and the hour of the day as regressors. The linear regression model can be described as

$$\begin{aligned} y_{t+h} =& \beta_0 + \beta_1 S_{t+h} + \beta_2 \left(S_{t+h}\right)^2 + \beta_3 \left(S_{t+h}\right)^3 + \beta_4 \cos\left(\frac{2\pi}{24} H_{t+h}\right) + \\ & \beta_5 \sin\left(\frac{2\pi}{24} H_{t+h}\right) + \beta_6 \cos\left(\frac{4\pi}{24} H_{t+h}\right) + \beta_7 \sin\left(\frac{4\pi}{24} H_{t+h}\right) + \varepsilon_{t+h}, \end{aligned} \tag{A.1}$$

where y_t is the dependent variable, which in this case is the wind power, S_{t+h} is the forecasted wind speed for the considered forecast horizon, H_{t+h} is the hour of the day for the considered forecast horizon, and $\beta_i, i = 0, \ldots, 7$ are the regression coefficients. As observed by Zhang and Wang [126], the sine and cosine terms account for diurnal periodicities through a Fourier series considering the time of the day. We fit and apply this alternative linear regression model in the same way as the linear regression model described in Section 3.3.3.

Neural Network as an Alternative Non-Linear Forecasting Model We select a simple feed forward neural network for our evaluation and test multiple configurations before choosing a configuration with two hidden layers of 10 and 7 neurons respectively. This network architecture is selected because it is the simplest we found that still returns accurate forecasts. Given this architecture, we train the neural network with the resilient backpropagation algorithm. The chosen activation function is a hyperbolic tangent and the input features are the same as those selected for the linear regression and random forest models (see Section 3.3.3). The parameters (i. e. weights) are fitted using the actual historical weather data and each ensemble member is passed through the network to get an ensemble of wind power forecasts.

Considering Different Distributions for Wind Speed and Wind Power From the literature, we identify both the truncated normal distribution and gamma distribution as possible candidates for modelling wind speed and wind power. To ensure robust results we consider both distributions and report the Continuous Ranked Probability Score (CRPS) values when using different combinations of the two distributions here. Following Gneiting [93], we implement a gamma distribution using $\mathrm{Gamma}(\alpha, \beta) = F_{\alpha,\beta}$, with $\alpha = a + b_1 x_1 + \ldots + b_M x_M$ and $\beta = c + d m_{\mathrm{ENS}}$ with $m_{\mathrm{ENS}} = a + b \sum_{m=1}^{M} x_m$.

Results The following tables (Table A.1 - Table A.4) show that the additional models confirm the results obtained with the linear regression and random forest forecasting models. They again highlight that post-processing the final wind power ensemble is crucial. Both the *One Step-T* and *Two Step-WT* strategies, which post-process the wind power ensemble, improve performance on the benchmark and the Swedish data sets. However, only post-processing the weather ensembles does not necessarily lead to improvements. Furthermore, the tables show no noticeable performance difference when using different distributions. We also report the results of models that do not include weather forecasts as input and are based solely on historical wind power values.

Table A.1.: Summary of mean CRPS on the test data for the onshore benchmark for all forecast horizons. The distribution used for post-processing the wind speed (W) and wind power (P) is shown in brackets. We compare linear regression (LR), alternative linear regression (ALR), random forest (RF), and a neural network (NN).

Strategy	6h	12h	18h	24h
LR No Weather	7.24	11.38	11.42	10.97
LR Raw	3.91	5.60	5.67	6.08
LR *One Step-T* (Gamma)	3.42	4.68	5.60	5.48
LR *One Step-T* (T-Normal)	3.53	4.88	5.71	5.76
LR *One Step-W* (Gamma)	3.80	5.29	5.66	5.96
LR *One Step-W* (T-Normal)	3.81	5.26	5.51	5.90
LR *Two Step-WT* (W: Gamma, T: Gamma)	3.52	4.96	5.63	5.78
LR *Two Step-WT* (W: Gamma, T: T-Normal)	3.42	4.96	5.61	5.72
LR *Two Step-WT* (W: T-Normal, T: Gamma)	3.50	4.87	5.64	5.84
LR *Two Step-WT* (W: T-Normal, T: T-Normal)	3.48	4.97	5.68	5.77
ALR No Weather	13.45	15.73	12.45	11.46
ALR Raw	4.59	5.79	6.21	5.93
ALR *One Step-T* (Gamma)	3.67	4.12	5.78	5.43
ALR *One Step-T* (T-Normal)	3.81	4.35	5.91	5.53
ALR *One Step-W* (Gamma)	4.73	4.32	6.31	5.82
ALR *One Step-W* (T-Normal)	4.73	4.19	6.38	5.91
ALR *Two Step-WT* (W: Gamma, T: Gamma)	4.01	4.22	5.86	5.52
ALR *Two Step-WT* (W: Gamma, T: T-Normal)	3.99	4.22	5.99	5.63
ALR *Two Step-WT* (W: T-Normal, T: Gamma)	3.98	3.93	5.89	5.56
ALR *Two Step-WT* (W: T-Normal, T: T-Normal)	3.85	4.04	6.15	5.89
RF No Weather	9.52	12.18	10.41	10.31
RF Raw	3.98	5.34	5.68	5.84
RF *One Step-T* (Gamma)	3.31	4.21	5.67	5.23
RF *One Step-T* (T-Normal)	3.61	4.63	5.84	5.38
RF *One Step-W* (Gamma)	3.85	4.44	5.82	5.83
RF *One Step-W* (T-Normal)	3.97	4.53	5.78	5.83
RF *Two Step-WT* (W: Gamma, T: Gamma)	3.34	4.08	5.73	5.45
RF *Two Step-WT* (W: Gamma, T: T-Normal)	3.57	4.30	5.86	5.52
RF *Two Step-WT* (W: T-Normal, T: Gamma)	3.41	4.05	5.73	5.42
RF *Two Step-WT* (W: T-Normal, T: T-Normal)	3.67	4.34	5.84	5.70
NN No Weather	7.80	11.34	10.15	10.31
NN Raw	7.16	10.42	10.19	10.19
NN *One Step-T* (Gamma)	6.05	8.57	8.56	7.83
NN *One Step-T* (T-Normal)	6.19	9.12	8.56	8.02
NN *One Step-W* (Gamma)	5.85	9.49	8.96	9.43
NN *One Step-W* (T-Normal)	5.90	9.60	9.38	9.44
NN *Two Step-WT* (W: Gamma, T: Gamma)	5.46	8.61	8.02	8.01
NN *Two Step-WT* (W: Gamma, T: T-Normal)	5.54	9.00	7.71	7.76
NN *Two Step-WT* (W: T-Normal, T: Gamma)	5.53	8.76	8.21	7.89
NN *Two Step-WT* (W: T-Normal, T: T-Normal)	5.44	8.99	7.71	7.65

Table A.2.: Summary of mean CRPS on the test data for the offshore benchmark for all forecast horizons. The distribution used for post-processing the wind speed (W) and wind power (P) is shown in brackets. We compare linear regression (LR), alternative linear regression (ALR), random forest (RF), and a neural network (NN).

Strategy	6h	12h	18h	24h
LR No Weather	35.64	54.01	66.41	68.15
LR Raw	18.98	23.51	25.46	24.11
LR *One Step-T* (Gamma)	17.97	22.52	24.95	23.69
LR *One Step-T* (T-Normal)	17.92	22.23	24.85	23.44
LR *One Step-W* (Gamma)	18.33	23.54	24.15	23.81
LR *One Step-W* (T-Normal)	18.74	23.31	24.51	23.88
LR *Two Step-WT* (W: Gamma, T: Gamma)	17.64	22.98	23.84	24.76
LR *Two Step-WT* (W: Gamma, T: T-Normal)	17.21	22.36	24.03	23.38
LR *Two Step-WT* (W: T-Normal, T: Gamma)	18.19	22.67	24.10	25.10
LR *Two Step-WT* (W: T-Normal, T: T-Normal)	17.30	22.08	24.28	23.49
ALR No Weather	69.48	70.90	72.09	69.61
ALR Raw	21.25	24.25	26.25	24.60
ALR *One Step-T* (Gamma)	19.94	23.48	25.86	24.68
ALR *One Step-T* (T-Normal)	19.85	23.12	25.34	25.10
ALR *One Step-W* (Gamma)	20.97	24.12	25.93	23.88
ALR *One Step-W* (T-Normal)	21.24	23.93	26.49	24.65
ALR *Two Step-WT* (W: Gamma, T: Gamma)	20.19	24.21	26.42	25.79
ALR *Two Step-WT* (W: Gamma, T: T-Normal)	20.22	23.67	25.29	24.94
ALR *Two Step-WT* (W: T-Normal, T: Gamma)	20.48	23.81	27.14	26.40
ALR *Two Step-WT* (W: T-Normal, T: T-Normal)	20.96	24.13	26.09	25.92
RF No Weather	51.37	58.67	64.36	64.75
RF Raw	20.77	23.80	25.00	25.54
RF *One Step-T* (Gamma)	18.91	21.97	25.03	24.54
RF *One Step-T* (T-Normal)	19.47	22.15	24.24	23.83
RF *One Step-W* (Gamma)	21.26	23.20	25.07	24.79
RF *One Step-W* (T-Normal)	21.45	23.22	25.41	24.93
RF *Two Step-WT* (W: Gamma, T: Gamma)	20.31	21.94	25.23	25.26
RF *Two Step-WT* (W: Gamma, T: T-Normal)	20.24	22.27	24.49	23.64
RF *Two Step-WT* (W: T-Normal, T: Gamma)	20.33	22.51	25.67	25.13
RF *Two Step-WT* (W: T-Normal, T: T-Normal)	20.47	22.43	25.56	24.08
NN No Weather	37.99	54.47	64.61	68.55
NN Raw	42.61	55.57	49.78	47.01
NN *One Step-T* (Gamma)	38.18	46.36	46.47	42.79
NN *One Step-T* (T-Normal)	38.19	45.67	44.03	43.71
NN *One Step-W* (Gamma)	37.80	49.98	44.11	41.22
NN *One Step-W* (T-Normal)	38.58	51.33	45.52	42.19
NN *Two Step-WT* (W: Gamma, T: Gamma)	36.89	43.32	42.87	39.69
NN *Two Step-WT* (W: Gamma, T: T-Normal)	36.07	42.74	40.47	38.90
NN *Two Step-WT* (W: T-Normal, T: Gamma)	38.13	43.60	44.01	40.38
NN *Two Step-WT* (W: T-Normal, T: T-Normal)	38.12	43.20	42.24	39.70

Table A.3.: Summary of mean CRPS on the test data for the use case of bidding zone 3 in Sweden for all forecast horizons. The distribution used for post-processing the wind speed (W) and wind power (P) is shown in brackets. We compare linear regression (LR), alternative linear regression (ALR), random forest (RF), and a neural network (NN).

Strategy	3h	6h	9h	12h	15h	18h	21h	24h
LR No Weather	68.25	125.24	169.95	206.38	228.61	244.21	255.13	273.54
LR Raw	57.84	78.46	81.03	79.75	86.20	89.05	89.06	93.33
LR *One Step-T* (Gamma)	45.87	55.41	62.07	67.34	70.53	64.36	67.97	67.45
LR *One Step-T* (T-Normal)	47.30	55.99	62.01	67.39	67.31	64.36	67.17	68.00
LR *One Step-W* (Gamma)	63.98	86.41	89.92	87.47	95.43	98.89	97.16	98.64
LR *One Step-W* (T-Normal)	64.51	86.61	90.05	87.27	96.89	99.55	97.95	98.99
LR *Two Step-WT* (W: Gamma, T: Gamma)	46.03	55.53	62.90	67.13	69.74	64.54	67.28	67.65
LR *Two Step-WT* (W: Gamma, T: T-Normal)	46.42	54.69	61.06	66.47	67.40	64.72	65.97	65.77
LR *Two Step-WT* (W: T-Normal, T: Gamma)	46.73	54.88	62.44	66.65	69.04	62.50	68.34	66.76
LR *Two Step-WT* (W: T-Normal, T: T-Normal)	47.08	55.32	62.04	65.56	66.86	63.70	67.05	65.83
ALR No Weather	276.01	282.10	280.85	287.40	292.19	290.27	279.83	280.57
ALR Raw	82.32	82.86	76.98	70.64	73.19	81.16	82.95	83.68
ALR *One Step-T* (Gamma)	59.74	58.48	56.85	57.52	56.18	55.63	60.11	56.36
ALR *One Step-T* (T-Normal)	59.95	59.01	57.42	58.53	57.43	55.85	61.02	57.44
ALR *One Step-W* (Gamma)	97.98	93.85	86.63	80.77	84.45	90.63	91.10	89.64
ALR *One Step-W* (T-Normal)	98.29	94.21	86.78	80.51	85.04	89.73	90.44	89.02
ALR *Two Step-WT* (W: Gamma, T: Gamma)	57.43	56.73	56.79	55.95	56.55	55.19	60.48	58.29
ALR *Two Step-WT* (W: Gamma, T: T-Normal)	58.22	58.27	56.81	56.76	56.73	56.47	60.51	57.52
ALR *Two Step-WT* (W: T-Normal, T: Gamma)	57.62	57.29	55.46	55.96	55.87	55.07	60.19	56.55
ALR *Two Step-WT* (W: T-Normal, T: T-Normal)	58.74	58.99	56.67	57.58	57.48	56.41	60.88	57.72
RF No Weather	170.90	192.13	211.51	235.68	252.47	252.25	251.61	260.93
RF Raw	60.71	71.31	69.72	67.74	73.39	82.01	83.67	88.09
RF *One Step-T* (Gamma)	43.54	48.30	52.78	55.29	54.46	52.92	57.36	57.11
RF *One Step-T* (T-Normal)	46.08	50.28	55.09	55.96	55.17	53.44	57.83	58.86
RF *One Step-W* (Gamma)	70.05	79.57	78.28	76.32	82.51	91.18	91.08	92.40
RF *One Step-W* (T-Normal)	69.96	79.95	78.21	76.50	82.77	90.38	91.47	91.15
RF *Two Step-WT* (W: Gamma, T: Gamma)	43.00	48.10	53.28	54.17	54.95	53.46	56.25	57.52
RF *Two Step-WT* (W: Gamma, T: T-Normal)	44.84	49.53	55.36	54.19	54.58	54.02	56.75	56.08
RF *Two Step-WT* (W: T-Normal, T: Gamma)	42.83	48.83	53.37	54.44	54.34	52.51	55.76	55.73
RF *Two Step-WT* (W: T-Normal, T: T-Normal)	45.20	49.92	55.48	54.27	55.36	53.95	57.34	57.24
NN No Weather	60.11	111.15	150.67	185.20	202.59	210.39	224.67	233.61
NN Raw	54.05	78.40	70.00	76.04	100.56	73.69	73.16	75.79
NN *One Step-T* (Gamma)	48.26	61.59	58.07	63.21	76.20	61.80	67.13	67.65
NN *One Step-T* (T-Normal)	48.38	64.54	59.89	66.47	78.67	64.66	69.14	68.11
NN *One Step-W* (Gamma)	56.59	78.18	72.74	71.19	98.05	72.66	72.87	73.20
NN *One Step-W* (T-Normal)	56.75	78.24	72.64	71.44	99.37	74.27	72.51	75.06
NN *Two Step-WT* (W: Gamma, T: Gamma)	48.31	61.04	59.85	63.69	76.67	63.02	69.13	67.87
NN *Two Step-WT* (W: Gamma, T: T-Normal)	48.56	64.67	60.48	64.27	77.11	63.77	68.66	66.32
NN *Two Step-WT* (W: T-Normal, T: Gamma)	48.09	62.30	60.30	63.38	75.97	63.69	68.40	67.69
NN *Two Step-WT* (W: T-Normal, T: T-Normal)	48.70	64.57	60.81	65.20	76.92	64.61	68.46	67.70

Table A.4.: Summary of mean CRPS on the test data for the use case of bidding zone 4 in Sweden for all forecast horizons. The distribution used for post-processing the wind speed (W) and wind power (P) is shown in brackets. We compare linear regression (LR), alternative linear regression (ALR), random forest (RF), and a neural network (NN).

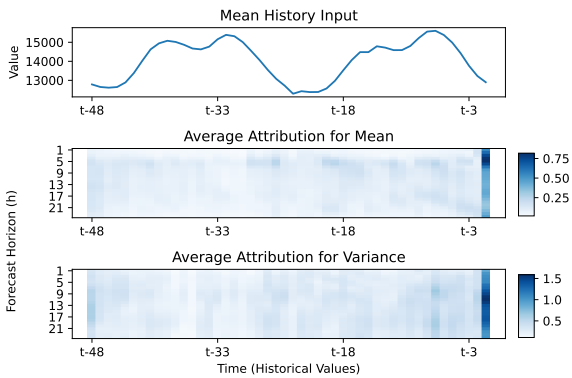
Strategy	3h	6h	9h	12h	15h	18h	21h	24h
LR No Weather	49.39	89.66	120.37	142.60	154.74	165.48	172.85	181.03
LR Raw	41.75	49.91	50.93	63.17	63.85	53.27	51.46	51.21
LR *One Step-T* (Gamma)	35.70	42.39	45.62	54.83	58.10	50.11	49.84	47.72
LR *One Step-T* (T-Normal)	36.17	41.60	43.15	49.68	53.51	47.86	45.92	43.69
LR *One Step-W* (Gamma)	41.78	51.18	52.17	57.57	61.95	52.73	51.68	49.12
LR *One Step-W* (T-Normal)	41.91	51.20	51.85	57.17	61.52	52.86	51.24	48.86
LR *Two Step-WT* (W: Gamma, T: Gamma)	35.44	42.47	45.94	54.84	58.20	50.14	49.22	47.74
LR *Two Step-WT* (W: Gamma, T: T-Normal)	36.08	40.73	42.56	49.52	53.96	48.05	45.62	43.49
LR *Two Step-WT* (W: T-Normal, T: Gamma)	35.31	42.24	45.74	54.04	57.33	49.50	49.64	46.95
LR *Two Step-WT* (W: T-Normal, T: T-Normal)	35.78	41.24	43.38	50.61	54.43	47.50	47.15	43.14
ALR No Weather	184.17	189.06	196.37	202.03	197.75	190.11	186.76	187.90
ALR Raw	42.92	44.07	42.82	54.38	55.16	44.77	40.33	38.30
ALR *One Step-T* (Gamma)	39.76	40.06	40.17	48.42	50.34	43.93	41.02	37.65
ALR *One Step-T* (T-Normal)	39.68	40.71	40.93	48.11	51.96	44.59	41.61	38.13
ALR *One Step-W* (Gamma)	42.95	46.43	44.15	51.19	55.29	45.58	40.60	38.03
ALR *One Step-W* (T-Normal)	43.37	46.75	44.18	50.98	55.58	45.98	41.08	38.17
ALR *Two Step-WT* (W: Gamma, T: Gamma)	38.16	40.60	39.50	48.42	50.72	43.77	41.47	38.36
ALR *Two Step-WT* (W: Gamma, T: T-Normal)	38.99	41.41	40.37	47.80	50.26	44.40	41.41	37.82
ALR *Two Step-WT* (W: T-Normal, T: Gamma)	38.91	40.44	39.50	48.32	50.54	43.47	41.41	37.58
ALR *Two Step-WT* (W: T-Normal, T: T-Normal)	39.15	40.42	40.11	48.34	51.39	44.39	41.73	37.90
RF No Weather	116.32	129.71	148.64	166.77	171.40	168.17	169.95	175.11
RF Raw	36.97	41.33	40.74	50.67	51.60	45.54	42.26	39.41
RF *One Step-T* (Gamma)	33.34	37.24	38.29	44.51	47.28	43.09	40.58	37.88
RF *One Step-T* (T-Normal)	34.33	38.25	39.30	44.60	48.02	44.28	41.72	38.18
RF *One Step-W* (Gamma)	36.63	42.63	41.89	46.57	51.59	45.69	43.07	38.57
RF *One Step-W* (T-Normal)	36.81	42.91	41.94	46.51	51.16	46.24	43.16	38.31
RF *Two Step-WT* (W: Gamma, T: Gamma)	32.47	37.03	37.82	43.25	48.01	43.12	41.11	37.42
RF *Two Step-WT* (W: Gamma, T: T-Normal)	33.71	37.91	38.76	43.33	47.90	44.73	41.72	37.32
RF *Two Step-WT* (W: T-Normal, T: Gamma)	32.44	37.31	37.87	43.74	46.81	43.13	40.73	37.25
RF *Two Step-WT* (W: T-Normal, T: T-Normal)	33.52	38.25	39.21	43.95	48.08	44.74	42.04	37.33
NN No Weather	49.75	90.51	121.89	150.22	157.56	163.37	168.79	174.76
NN Raw	47.52	48.68	49.41	58.64	58.55	61.21	55.49	51.71
NN *One Step-T* (Gamma)	38.92	44.02	45.03	57.58	54.49	49.75	47.89	44.22
NN *One Step-T* (T-Normal)	40.26	44.24	45.80	56.82	56.79	51.21	47.55	45.02
NN *One Step-W* (Gamma)	70.80	92.27	105.05	100.64	98.54	80.15	88.04	87.43
NN *One Step-W* (T-Normal)	51.76	68.62	77.67	78.28	74.47	61.77	65.74	65.53
NN *Two Step-WT* (W: Gamma, T: Gamma)	43.55	53.51	51.32	61.07	58.26	55.46	49.56	50.08
NN *Two Step-WT* (W: Gamma, T: T-Normal)	45.79	53.14	52.60	59.16	57.91	55.56	49.88	52.13
NN *Two Step-WT* (W: T-Normal, T: Gamma)	40.89	51.27	50.19	56.03	55.45	52.22	47.70	48.15
NN *Two Step-WT* (W: T-Normal, T: T-Normal)	43.45	50.78	49.57	56.19	54.98	52.43	48.17	49.43

B

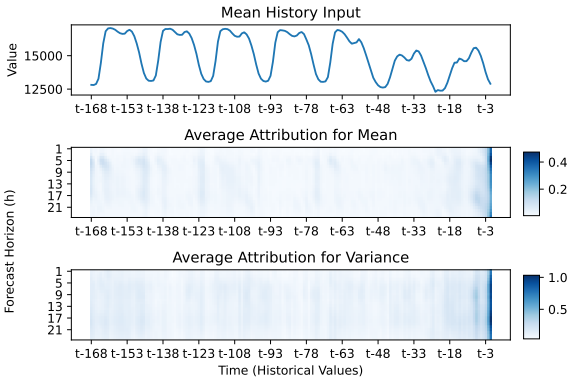
Explaining the Origins of Uncertainty in Probabilistic Forecasts

On the following pages, we present further results from Chapter 6. The further results consider temporally similar attributes for points in time not presented in Chapter 6. These extra results are as follows:

- Figure B.1 – Figure B.6 show the temporally similar absolute attributes for the history input for the remaining days of the week for the Sweden load data set. These days are Monday, Wednesday, Thursday, Friday, Saturday, and Sunday.
- Figure B.7 – Figure B.12 show the temporally similar absolute attributes for the history input for the remaining days of the week for the Sweden load data set. These days are Monday, Wednesday, Thursday, Friday, Saturday, and Sunday.
- Figure B.13 – Figure B.18 show the temporally similar absolute attributes for the history input for the remaining days of the week for the Price data set. These days are Monday, Wednesday, Thursday, Friday, Saturday, and Sunday.
- Figure B.19 – Figure B.24 show the temporally similar absolute attributes for the exogenous feature inputs for the remaining days of the week for the Price data set using the model considering $48\,\mathrm{h}$ of historical information. These days are Monday, Wednesday, Thursday, Friday, Saturday, and Sunday.
- Figure B.25 – Figure B.30 show the temporally similar absolute attributes for the exogenous feature inputs for the remaining days of the week for the Price data set using the model considering $168\,\mathrm{h}$ of historical information. These days are Monday, Wednesday, Thursday, Friday, Saturday, and Sunday.
- Figure B.31 shows the temporally similar attributions for the exogenous forecast features at midnight and 11 am for the Solar data for the model considering $168\,\mathrm{h}$ of historical information.
- Figure B.32 shows the Static Mean Average Attributions for each input feature on the Solar data set for the model considering $168\,\mathrm{h}$ of historical information.

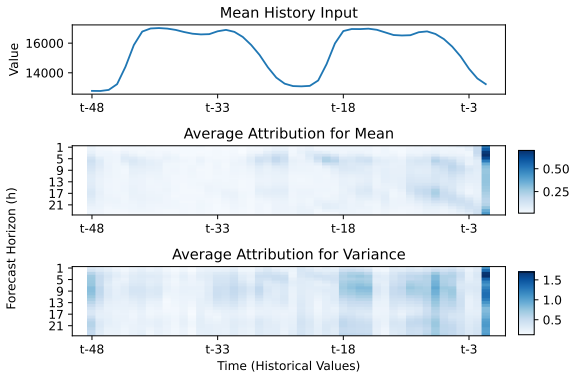


(a) 48h History Mean Monday

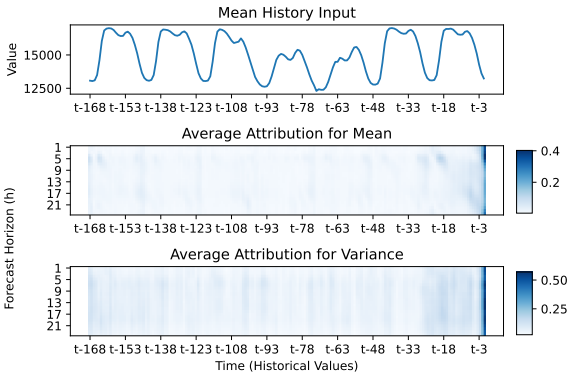


(b) 168h History Mean Monday

Figure B.1.: A comparison of the temporally similar attributions for Mondays on the Sweden load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

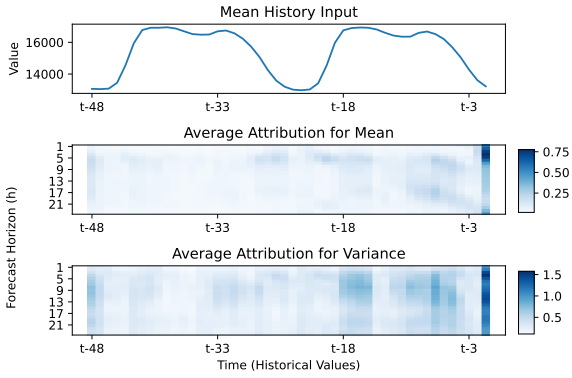


(a) 48h History Mean Wednesday

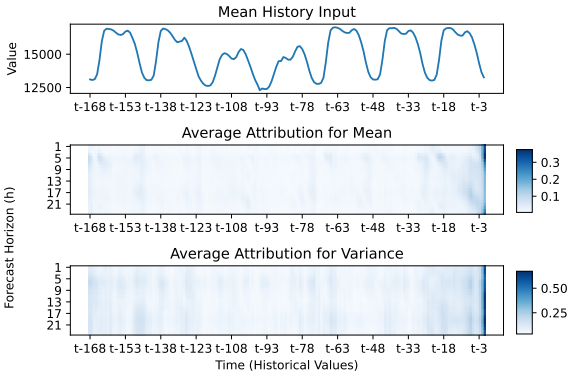


(b) 168h History Mean Wednesday

Figure B.2.: A comparison of the temporally similar attributions for Wednesdays on the Sweden load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

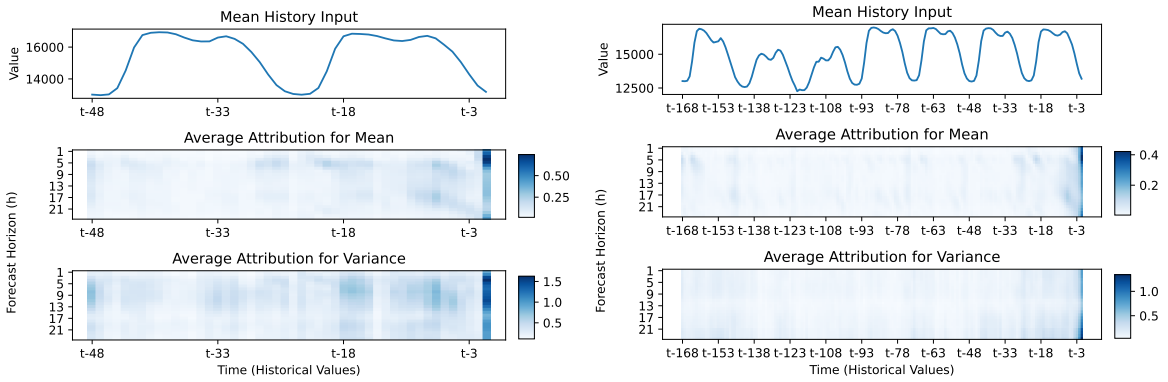


(a) 48h History Mean Thursday



(b) 168h History Mean Thursday

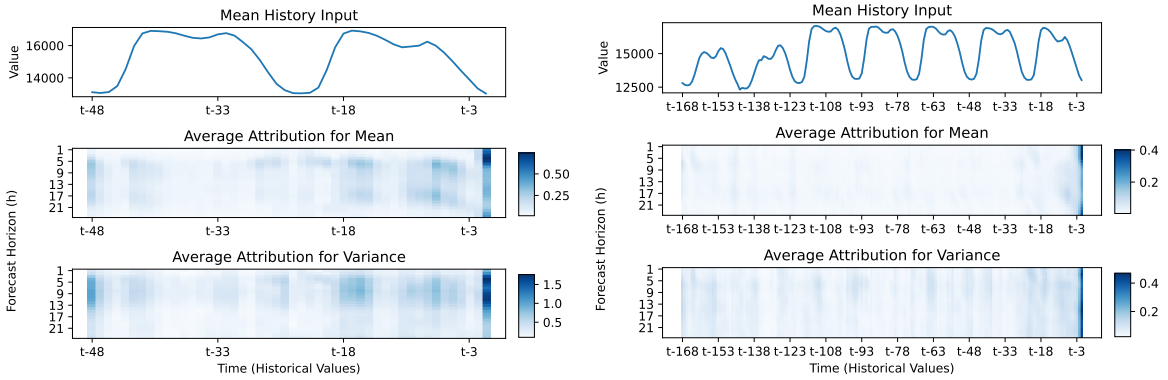
Figure B.3.: A comparison of the temporally similar attributions for Thursdays on the Sweden load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.



(a) 48h History Mean Friday

(b) 168h History Mean Friday

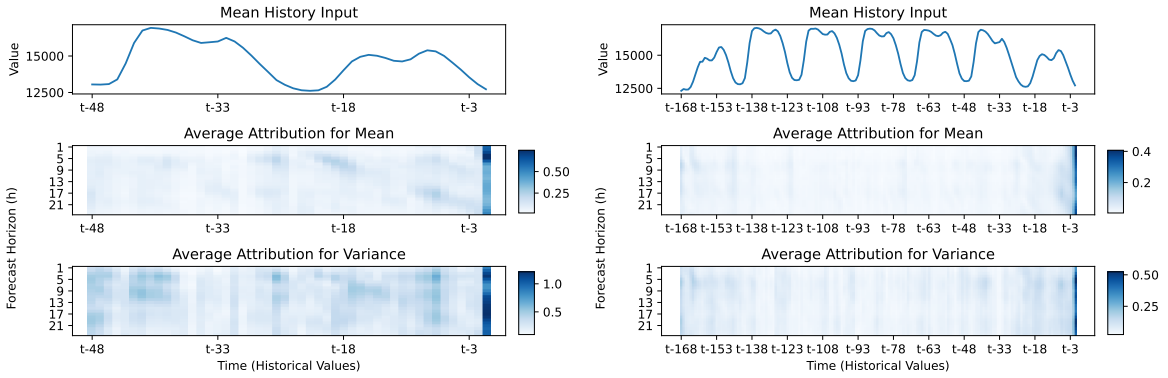
Figure B.4.: A comparison of the temporally similar attributions for Fridays on the Sweden load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.



(a) 48h History Mean Saturday

(b) 168h History Mean Saturday

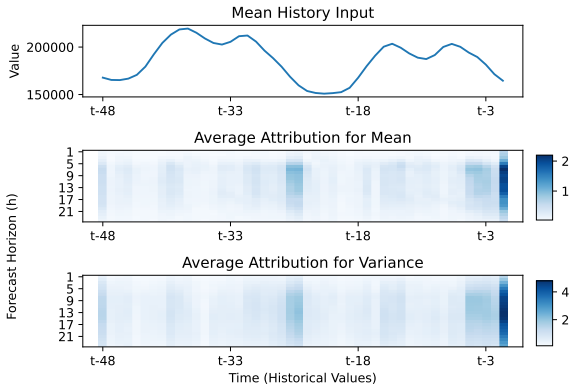
Figure B.5.: A comparison of the temporally similar attributions for Saturdays on the Sweden load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.



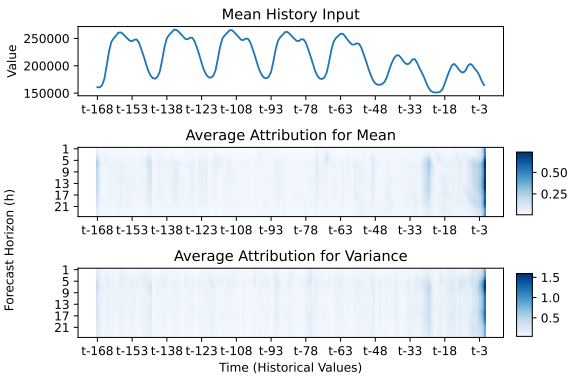
(a) 48h History Mean Sunday

(b) 168h History Mean Sunday

Figure B.6.: A comparison of the temporally similar attributions for Sundays on the Sweden load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

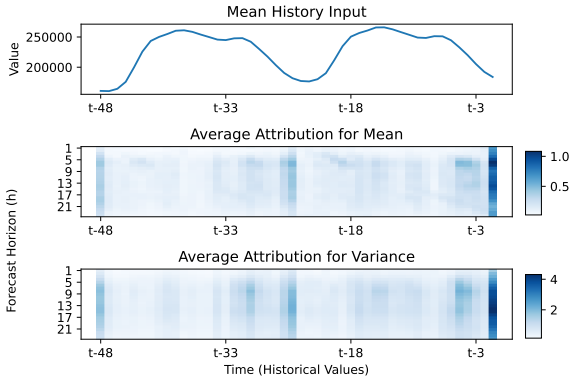


(a) 48h History Mean Monday

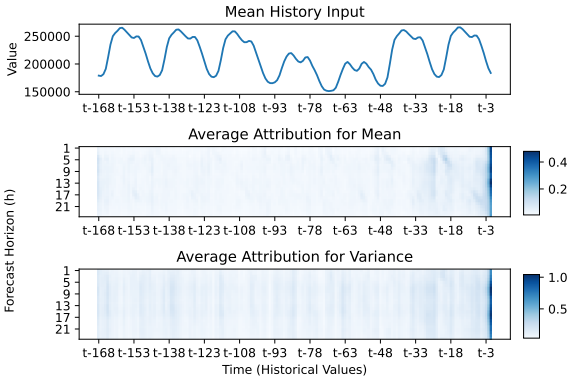


(b) 168h History Mean Monday

Figure B.7.: A comparison of the temporally similar attributions for Mondays on the Germany load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

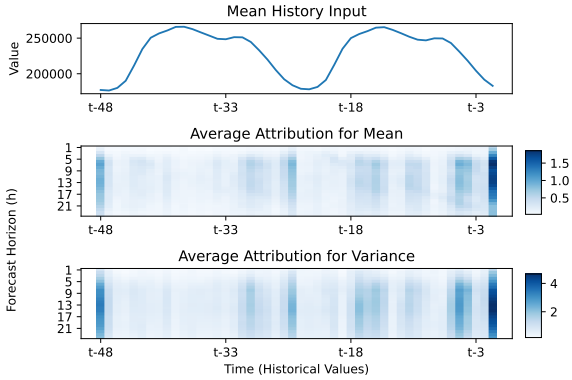


(a) 48h History Mean Wednesday

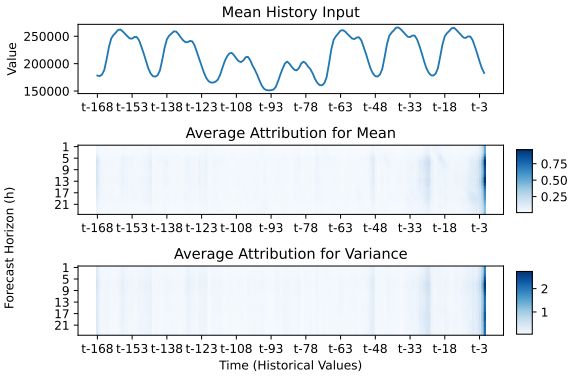


(b) 168h History Mean Wednesday

Figure B.8.: A comparison of the temporally similar attributions for Wednesdays on the Germany load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

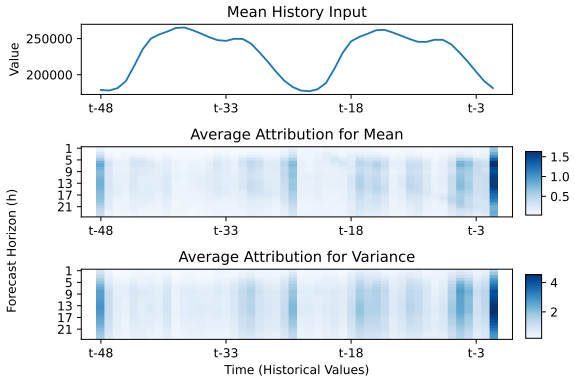


(a) 48h History Mean Thursday

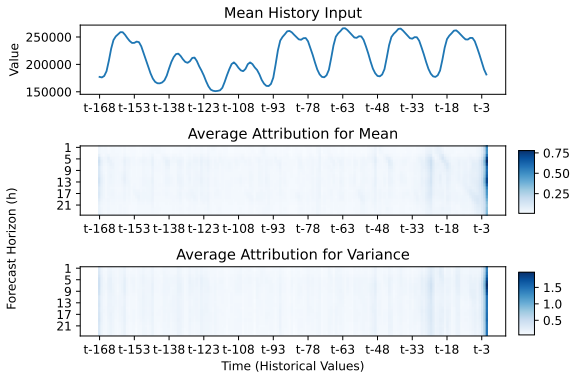


(b) 168h History Mean Thursday

Figure B.9.: A comparison of the temporally similar attributions for Thursdays on the Germany load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

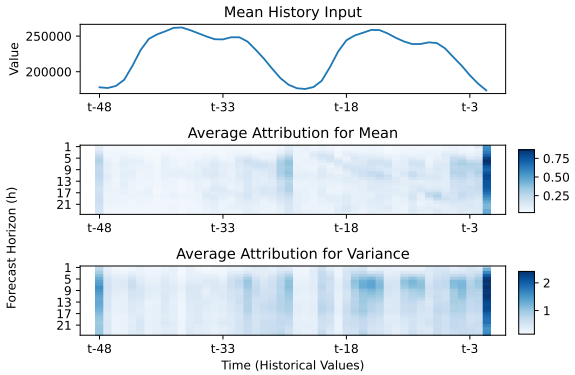


(a) 48h History Mean Friday

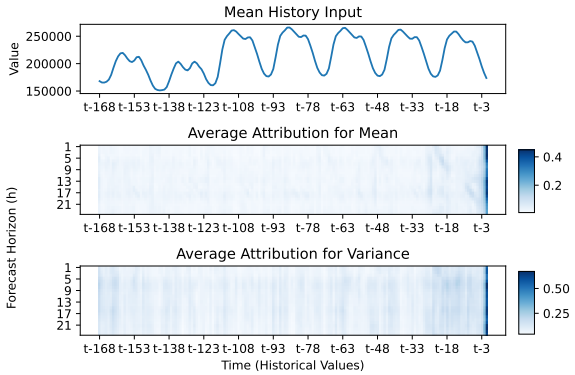


(b) 168h History Mean Friday

Figure B.10.: A comparison of the temporally similar attributions for Fridays on the Germany load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

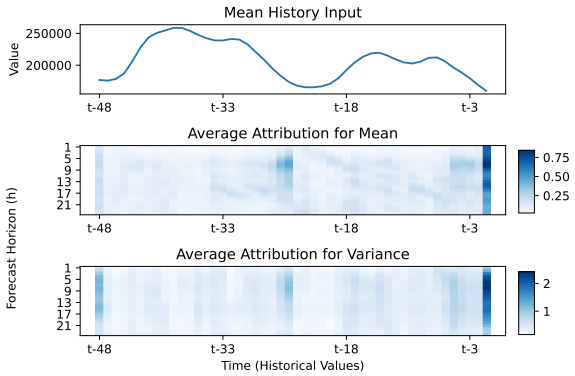


(a) 48h History Mean Saturday

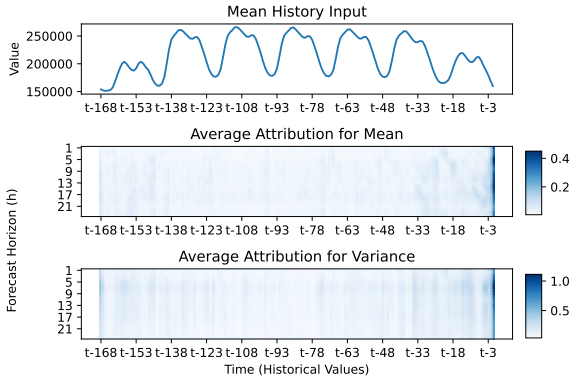


(b) 168h History Mean Saturday

Figure B.11.: A comparison of the temporally similar attributions for Saturdays on the Germany load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.

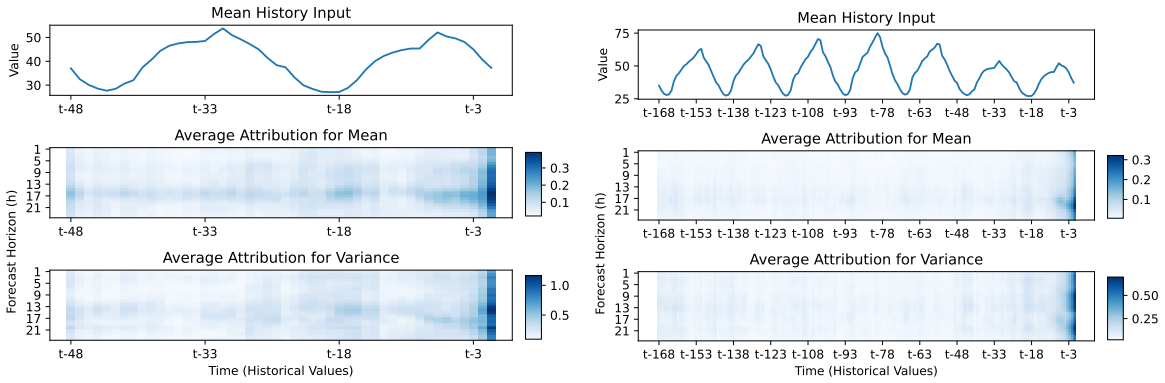


(a) 48h History Mean Sunday



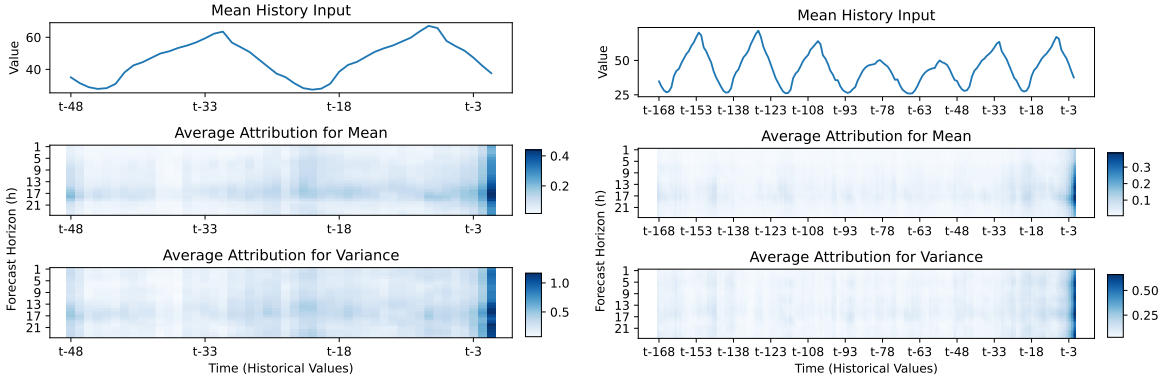
(b) 168h History Mean Sunday

Figure B.12.: A comparison of the temporally similar attributions for Sundays on the Germany load data. The models with 48 h History and 168 h History are compared. The neural network does not consider any exogenous features and always generates a probabilistic forecast for the next 24 h.



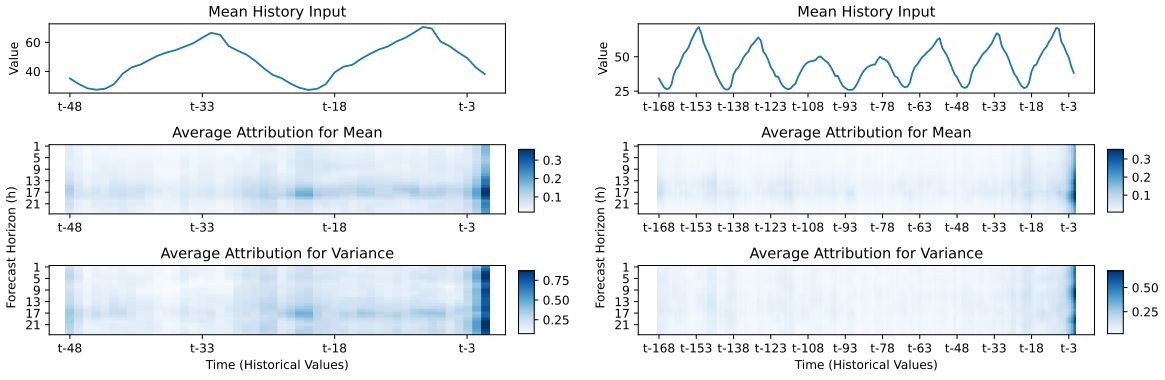
(a) 48h History **(b)** 168h History

Figure B.13.: A comparison of temporally similar absolute attributions for the history input on Mondays between the model with $48\,\mathrm{h}$ History and the model with $168\,\mathrm{h}$ History on the Price data. The neural network also considers exogenous features as an input and always generates a probabilistic forecast for the next $24\,\mathrm{h}$.



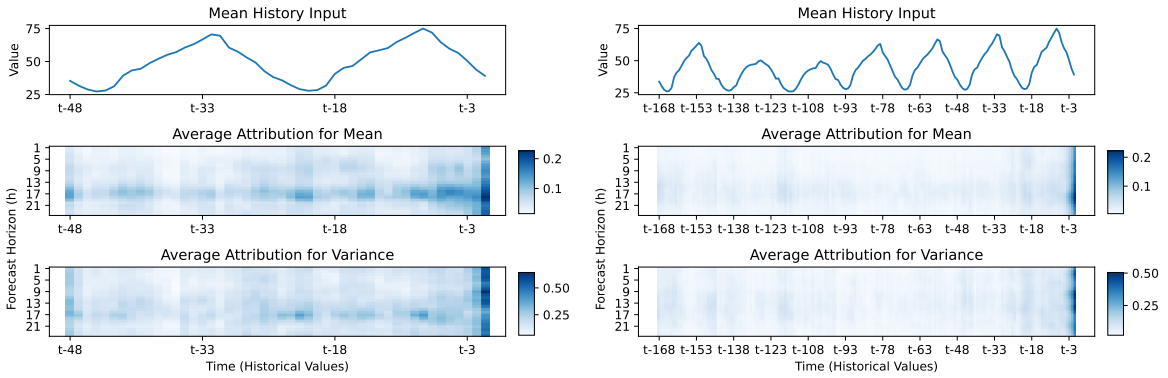
(a) 48h History **(b)** 168h History

Figure B.14.: A comparison of temporally similar absolute attributions for the history input on Wednesdays between the model with $48\,\mathrm{h}$ History and the model with $168\,\mathrm{h}$ History on the Price data. The neural network also considers exogenous features as an input and always generates a probabilistic forecast for the next $24\,\mathrm{h}$.



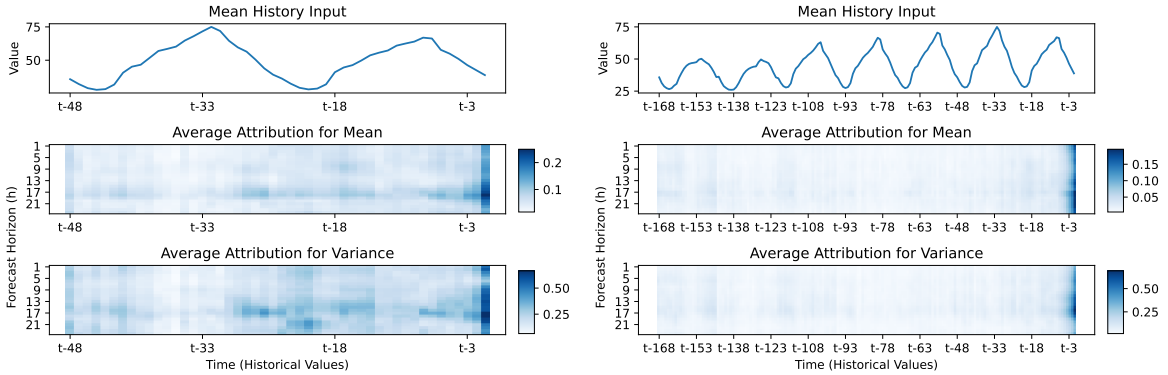
(a) 48h History **(b)** 168h History

Figure B.15.: A comparison of temporally similar absolute attributions for the history input on Thursdays between the model with $48\,\mathrm{h}$ History and the model with $168\,\mathrm{h}$ History on the Price data. The neural network also considers exogenous features as an input and always generates a probabilistic forecast for the next $24\,\mathrm{h}$.



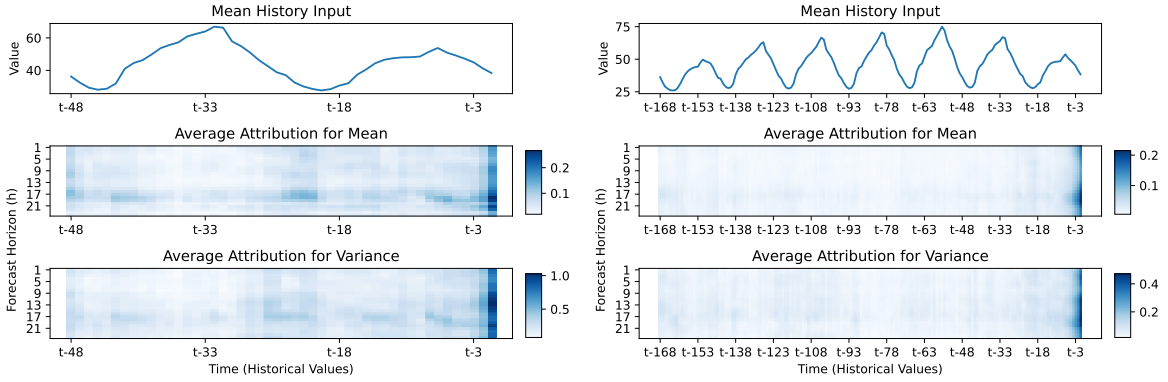
(a) 48h History **(b)** 168h History

Figure B.16.: A comparison of temporally similar absolute attributions for the history input on Fridays between the model with $48\,\mathrm{h}$ History and the model with $168\,\mathrm{h}$ History on the Price data. The neural network also considers exogenous features as an input and always generates a probabilistic forecast for the next $24\,\mathrm{h}$.



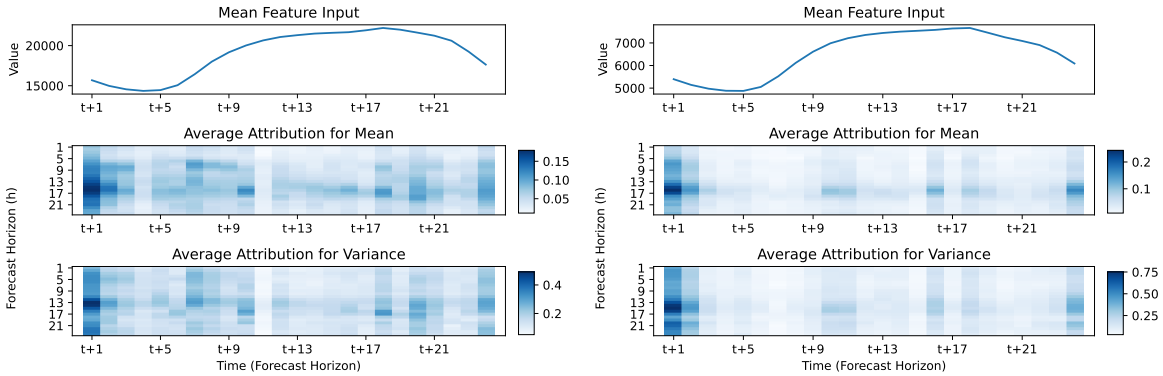
(a) 48h History **(b)** 168h History

Figure B.17.: A comparison of temporally similar absolute attributions for the history input on Saturdays between the model with $48\,\mathrm{h}$ History and the model with $168\,\mathrm{h}$ History on the Price data. The neural network also considers exogenous features as an input and always generates a probabilistic forecast for the next $24\,\mathrm{h}$.



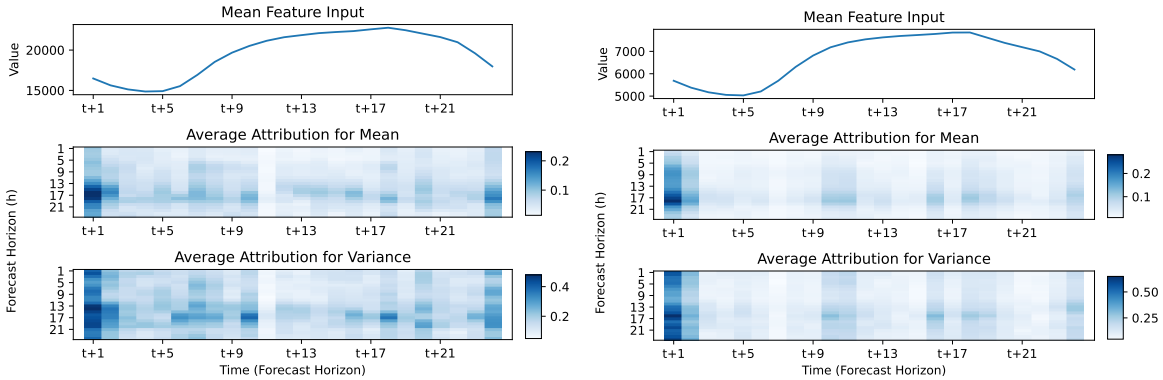
(a) 48h History **(b)** 168h History

Figure B.18.: A comparison of temporally similar absolute attributions for the history input on Sundays between the model with $48\,\mathrm{h}$ History and the model with $168\,\mathrm{h}$ History on the Price data. The neural network also considers exogenous features as an input and always generates a probabilistic forecast for the next $24\,\mathrm{h}$.



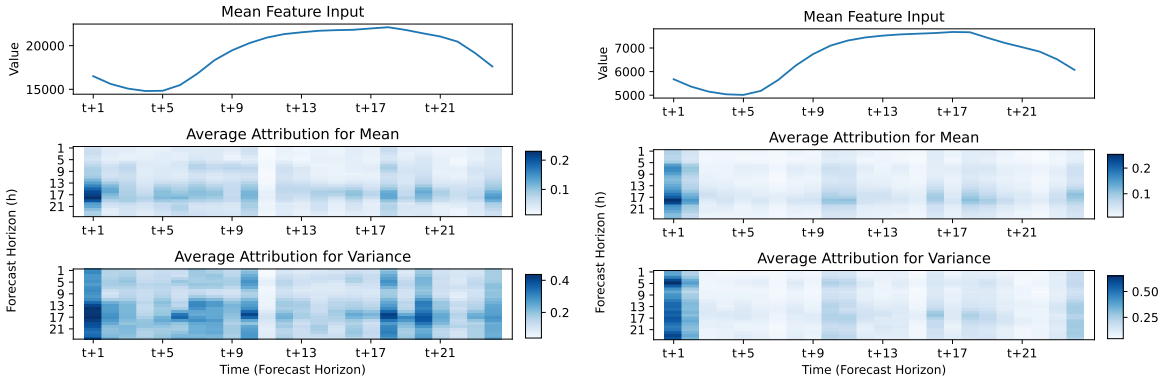
(a) Total Load **(b)** Zonal Load

Figure B.19.: A comparison of the temporally similar absolute attributes for Mondays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 48 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



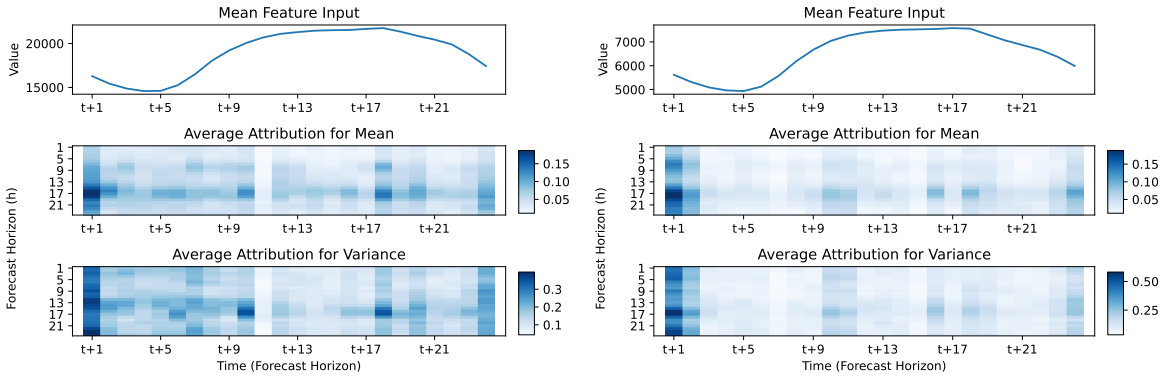
(a) Total Load **(b)** Zonal Load

Figure B.20.: A comparison of the temporally similar absolute attributes for Wednesdays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 48 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



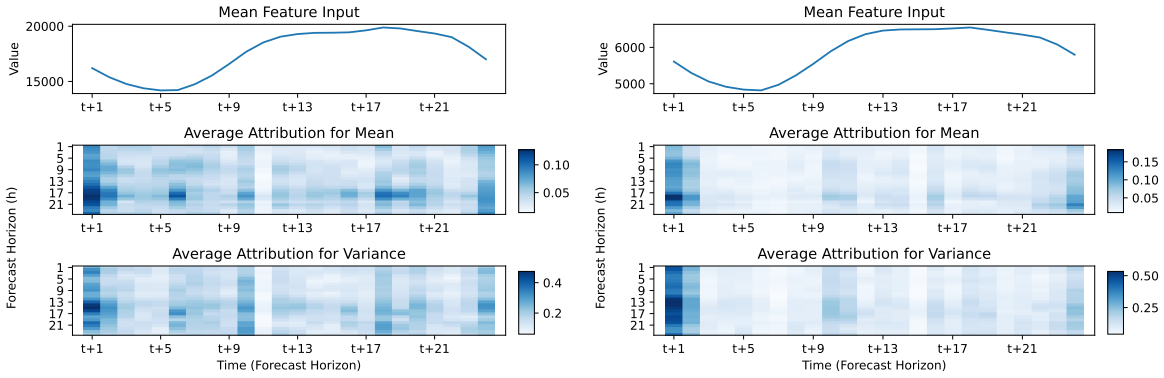
(a) Total Load **(b)** Zonal Load

Figure B.21.: A comparison of the temporally similar absolute attributes for Thursdays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 48 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



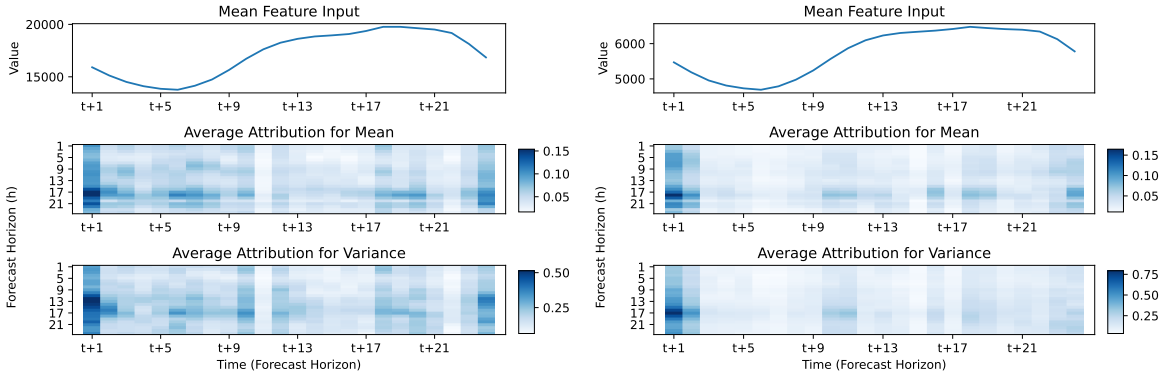
(a) Total Load **(b)** Zonal Load

Figure B.22.: A comparison of the temporally similar absolute attributes for Fridays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 48 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



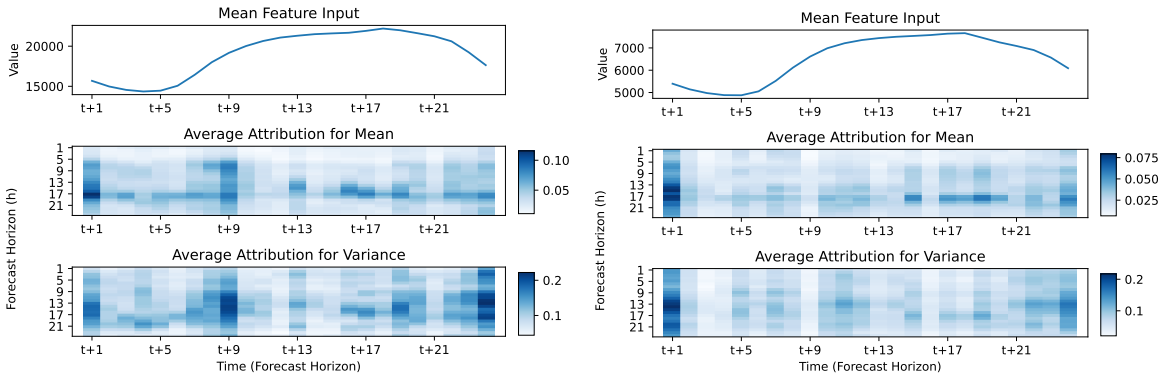
(a) Total Load **(b)** Zonal Load

Figure B.23.: A comparison of the temporally similar absolute attributes for Saturdays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 48 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



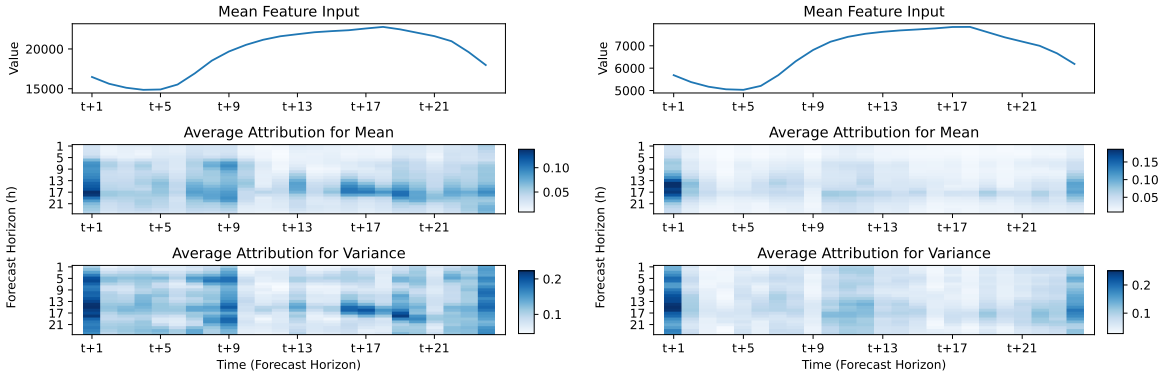
(a) Total Load **(b)** Zonal Load

Figure B.24.: A comparison of the temporally similar absolute attributes for Sundays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 48 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



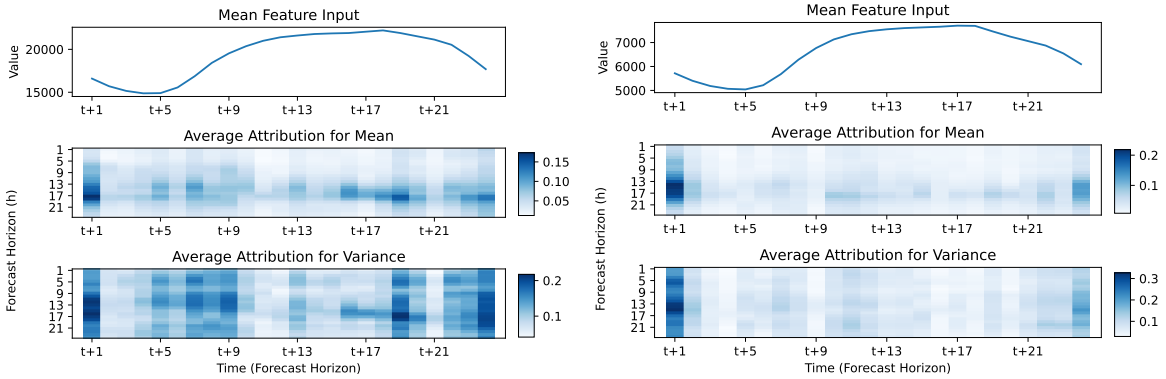
(a) Total Load **(b)** Zonal Load

Figure B.25.: A comparison of the temporally similar absolute attributes for Mondays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 168 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



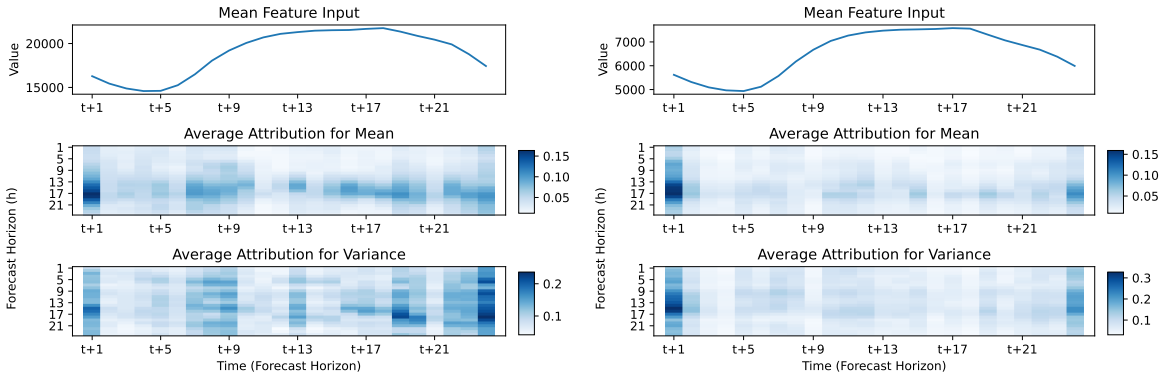
(a) Total Load **(b)** Zonal Load

Figure B.26.: A comparison of the temporally similar absolute attributes for Wednesdays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 168 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



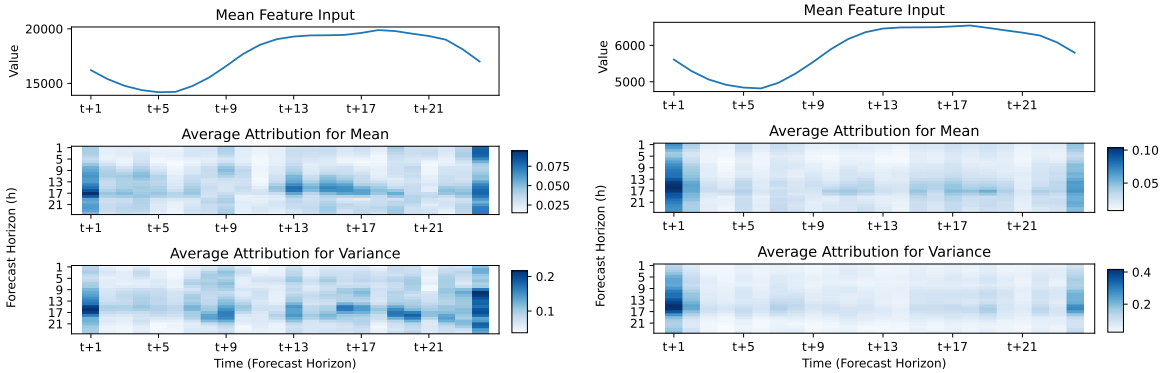
(a) Total Load **(b)** Zonal Load

Figure B.27.: A comparison of the temporally similar absolute attributes for Thursdays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 168 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



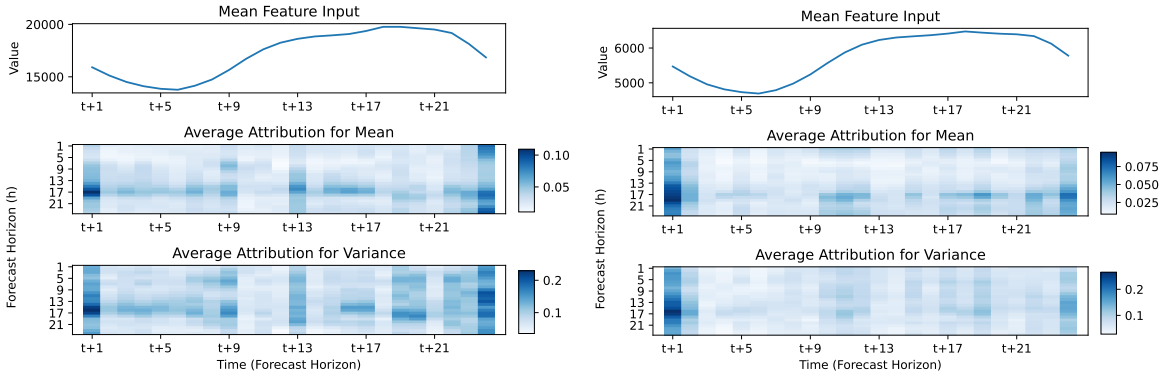
(a) Total Load **(b)** Zonal Load

Figure B.28.: A comparison of the temporally similar absolute attributes for Fridays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 168 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



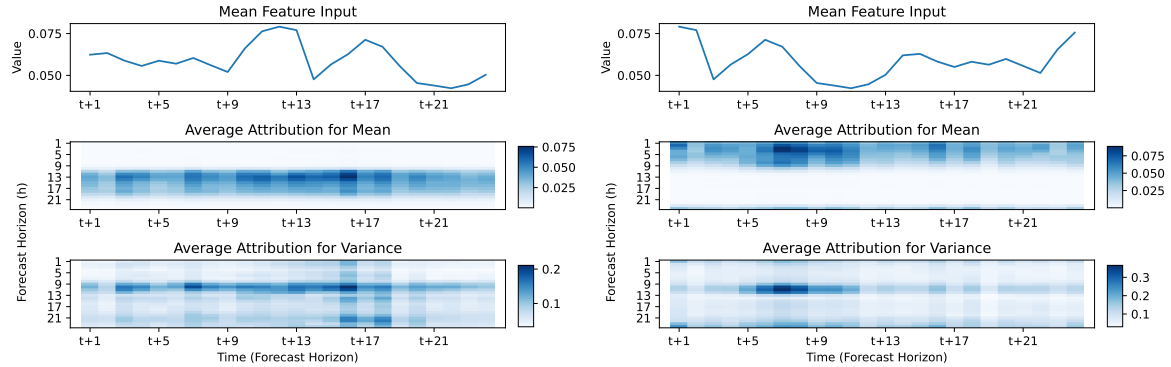
(a) Total Load **(b)** Zonal Load

Figure B.29.: A comparison of the temporally similar absolute attributes for Saturdays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 168 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



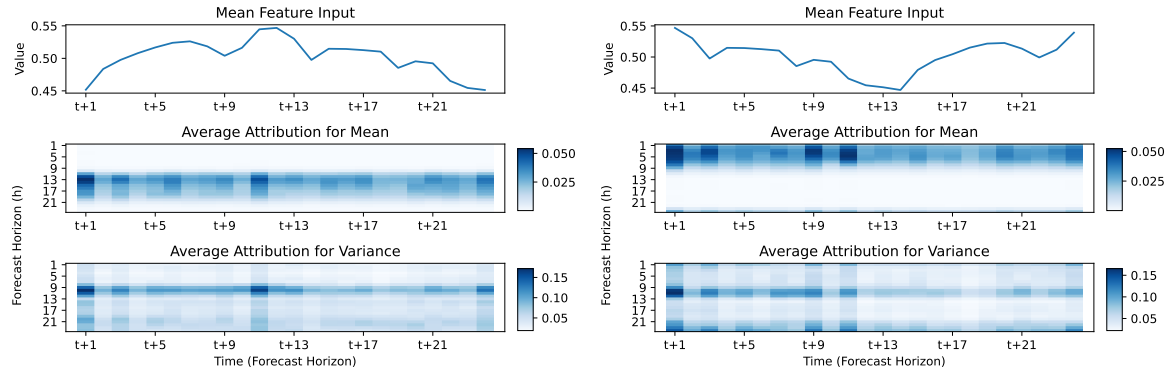
(a) Total Load **(b)** Zonal Load

Figure B.30.: A comparison of the temporally similar absolute attributes for Sundays for the two exogenous forecast features for the Price data set. The comparison is created using the model that considers 168 h of historical information as well as the exogenous features and generates a forecast for the next 24 h.



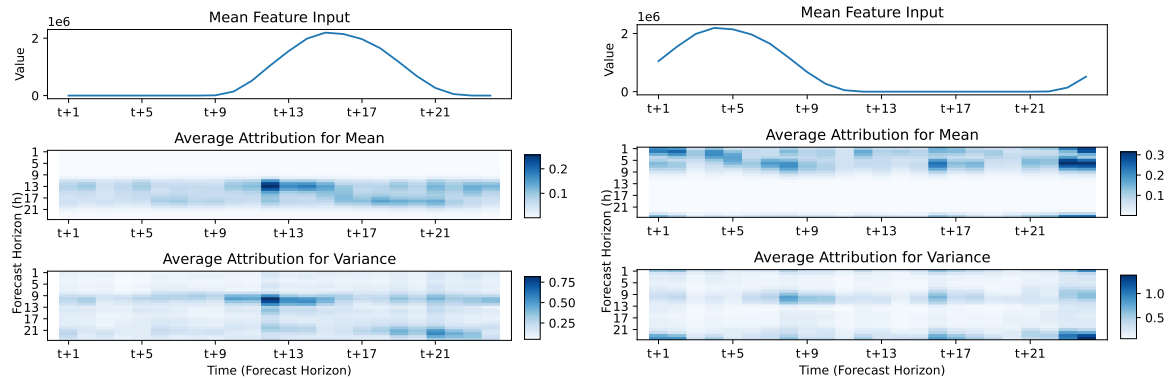
(a) TCLW,Midnight

(b) TCLW, 11 am



(c) TCC, Midnight

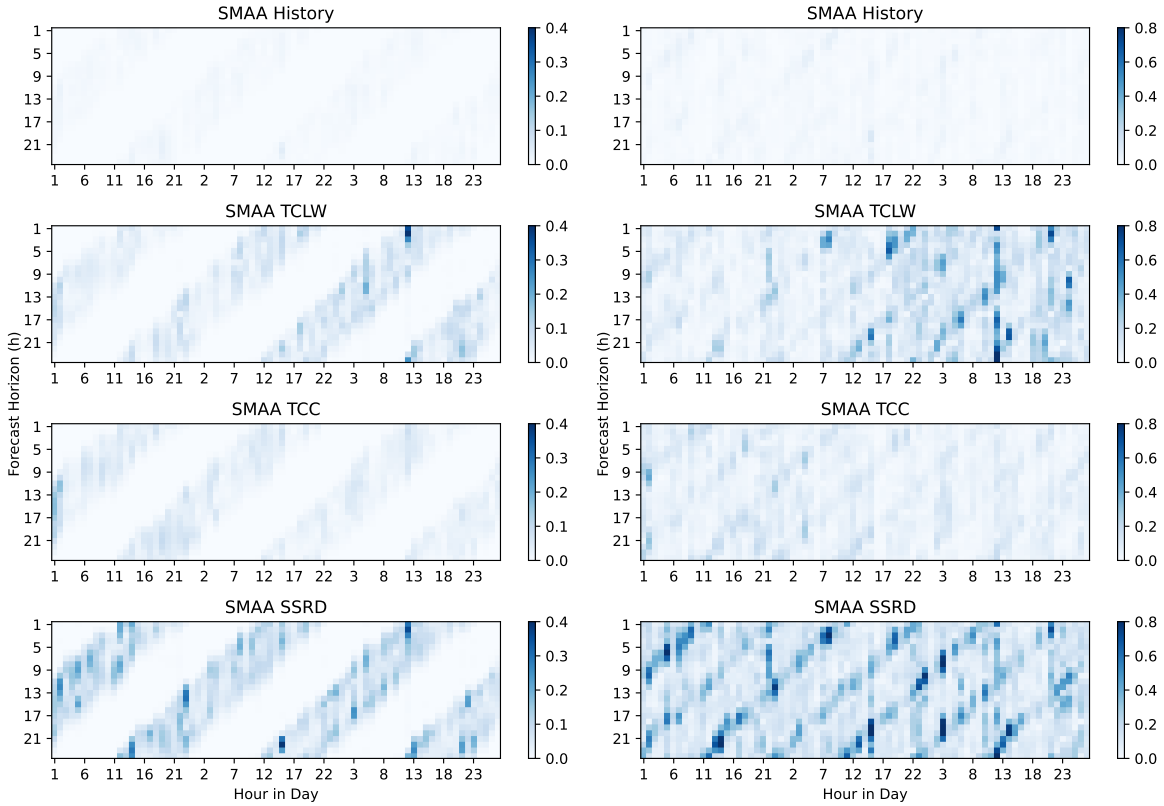
(d) TCC, 11 am



(e) SSRD,Midnight

(f) SSRD, 11 am

Figure B.31.: A comparison of the temporally similar absolute attributes at midnight and 11 am for the three exogenous forecast features for the Solar data set. The comparison is created using the model that considers $168\,\mathrm{h}$ of historical information as well as the exogenous features and generates a forecast for the next $24\,\mathrm{h}$.



(a) Attributions for the mean forecast

(b) Attributions for the variance forecast

Figure B.32.: The Static Mean Average Attributions (SMAA) for each input feature for the Solar data set using the model considering 168 h of historical information. The diagonal pattern for each feature indicates that only forecast horizons that occur during the day consider the features. Furthermore, the darker colours for SSRD indicates that this feature has the highest attributions on average and is therefore the most important feature for the forecast.

Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.