Exploring Probabilistic Short-Term Water Demand Forecasts at District Level Using Neural Networks

Christiaan Ruben Jacob Wewer

Delft University of Technology

TU Delft

Exploring Probabilistic Short-Term Water Demand Forecasts at District Level Using Neural Networks

by

Christiaan Ruben Jacob Wewer

in partial fulfilment of the requirements for the degree of
Master of Science
in Civil Engineering,
at Delft University of Technology

Thesis Committee:
Dr. ir. R. (Riccardo) Taormina (Chair)
Dr. R.R.P (Ronald) van Nooijen

TU Delft

Preface

Working on this thesis has been a challenging yet rewarding process, focused on exploring and addressing scientific questions. From designing experiments to tackling complex problems, this journey has been a valuable learning process that shaped my knowledge and skills. Along the way, I gained new insights and learned more about myself. While much of the work required focus and independence, I am thankful for the support of those around me. My family and friends provided encouragement and helped me stay grounded throughout this journey, and I am grateful for their presence.

I would also like to thank my supervisors, Riccardo Taormina and Ronald van Rooien, for their guidance, feedback and support throughout this process. A special thanks to Riccardo for the opportunity to present some early work at the European Geosciences Union (EGU) General Assembly, an experience that enriched my academic journey.

Christiaan Ruben Jacob Wewer
Delft, November 2024

Abstract

In a world with accelerating climate change, rapid population increase and urbanization, urban water systems are under a growing stress. Thus precise short- and medium-term water demand forecasts are needed to optimize water supply operations. Water demand is influenced by human behavior and industrial activities which bring uncertainty, hence it is useful to utilize probabilistic methods to forecast water demand. This thesis provides an overview of probabilistic methods to predict water demand 24 hours ahead, highlighting their advantages, disadvantages, the accuracy of their interval and point forecasts. The case study uses the dataset from the Battle of Water Demand 2024, covering two years and two months of data across 10 districts in Ferrara, Italy, including residential, hospital, countryside, city center, and industrial districts. Three commonly used probabilistic extensions of neural networks were applied: QR (Quantile Regression), MDN (Gaussian Mixture Density Network), and an adapted CQR (Conformal Quantile Regression) method with online updating.

First, forecast models were developed to obtain probabilistic predictions. Three neural network architectures were investigated: a linear model, an MLP (multi-layer perceptron), and an LSTM (Long Short-Term Memory) model, along with a seasonal moving average as a benchmark. These neural network models were trained per district and collectively across districts. When trained per district, the linear model performed most accurately. When trained together, the MLP model performed best, but the linear model generalized the best overall, with the MLP generalizing second-best. The LSTM model had the worst performance. In districts with less heteroscedasticity in the demand pattern, the benchmark model performed on par with the neural networks in the end of the forecasting horizon, indicating that complex models are not always necessary. A categorical variable to determine the DMA did not improve the point forecasts.

Because the MLP with solely lagged features ultimately had the best performance for point forecasts, this model was used for probabilistic extensions to estimate the 0.95 prediction interval. The MLP was extended with the aforementioned probabilistic extensions. The probabilistic models were assessed in terms of reliability with where a probability was computed that tells whether the 0.95 prediction interval is reached as well as sharpness which tells how wide the interval is. Finally the Winkler Score is used that computes a trade-off between both.

Both models that fully learn the prediction interval (QR and MDN) were more difficult to calibrate, and further research is needed to calibrate them accordingly. By training these models jointly on the 10 DMAs, the coverages did vary per DMA. This can potentially be solved by training one model per DMA or by using regularization per DMA to push the model to have a similar coverage per DMA.

The MCD model had difficulty to adapt the prediction interval over the forecasting horizon, causing the coverage to reduce. The QR and MDN models also have trade off imbalances between reliability and sharpness over the forecasting horizon on the testing set, which have more random patterns. Interestingly the Conformal Prediction algorithm maintains its coverage best over the forecasting horizon and increases the sharpness, which is due to the online updating procedure. When allowing small decreases up to 0.02 probability in coverage on the testing set, the CQR model performs best according to the Winkler Score. The MDN performs best when larger drops of coverage are allowed.

Analyzing the rolling coverage over time shows improvements are possible, especially in late spring and summer periods there is under-coverage. There is also still a difference of coverage between weekdays and weekends. This indicates there is still epistemic uncertainty left to reduce, which is the uncertainty of data and model parameters. More features are recommended to reduce this which are categorical features as well as future weather data. This is recommended to investigate for non-industrial DMAs, by assuming a perfect forecast. A larger dataset may also be beneficial to obtain better performing models.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
DMA	District Metered Area
MLP	Multi-Layer Perceptron
LSTM	Long-Short Term Memory
QR	Quantile Regression
MDN	Mixture Density Network
CQR	Conformal Quantile Regression
MCD	Monte-Carlo Dropout
MAE	Mean Absolute Error
RMAE	Relative Mean Absolute Error
MAPE	Mean Absolute Percentage Error
GR	Generalization Ratio
PICP	Prediction Interval Coverage Probability
CG	Coverage Gap
PINAW	Prediction Interval Normalized Average Width
WS	Winkler Score
CWS	Conditional Winkler Score
CNWS	Conditional Normalized Winkler Score
RQE	Relative Quantile Error

1. Introduction

Rapid population growth [38] leading to increased urbanization [18], along with the impacts of climate change, are expected to raise global water demands [55]. Consequently, urban water systems are experiencing increasing stress [23]. Precise water demand forecasting is therefore essential. This will help policymakers, advisors, and engineers of drinking water utilities optimize water supply operations, improve water treatment plants, detect bursts, and efficiently schedule pumps [17, 42, 4]. Because these operations require a forecasting horizon of at least 24 hours, the models developed in this thesis also focus on a forecasting horizon of 24 hours.

In this research, deep learning models were chosen for water demand forecasting due to their ability in handling non-linear complex data and flexible model design. This flexibility allows for experimenting with different architectures and combining them with probabilistic methods to extend the model. Furthermore it is known that in different disciplines more data improves the deep learning models performance and generalization, thus it makes sense to explore if training a model on one district or multiple districts has an impact. Within these aspects, different models were investigated, and the best-performing model was extended for probabilistic forecasting. This leads to the first two research questions:

RQ1: Which deterministic deep learning model is most suitable for short term water demand point predictions?

RQ2: How does the accuracy of short-term water demand forecasting differ between individual models for each district metered area and a single model for all areas?

Of these models the inputs were selected to be the past week of hourly lagged water demand. For the models trained on an all the districts a version was added that incorporates an indicator of the district.

In water demand forecasting literature, many models are still deterministic (for example [24] and [5] and more studies can be found in the literature review in chapter 2). This means that when a certain demand is predicted, there is no notion of uncertainty. Unlike point forecasts, interval forecasts, a specific case of probabilistic forecasts, are capable of representing the inherent uncertainty in estimating an unrealized value. They provide upper and lower bounds, creating a range where the actual value is expected to fall, based on a specified probability. There are several methods to determine these bounds, many of which rely on the distribution of the target value or the errors from point forecasts.

It makes sense to use probabilistic forecasts to improve decision-making by having the interval size reflecting the confidence decision-makers can place in the predictions. Larger intervals imply lower forecast reliability to capture the process, while smaller intervals suggest higher reliability. This may guide adjustments in operational strategies, by improving the control of the aforementioned urban water operations. It is difficult to know beforehand which probabilistic methods will perform best, thus in this research a selection is made of four methods, of which three are often used. These are Monte-Carlo Dropout [20], Neural Quantile Regression [11], Gaussian Mixture Density Network [9, 22] and an adapted version of Conformalized Quantile Regression [45, 29]. The last method is a combination between two conformal prediction methods, modified for multi-step use. This leads to the following research question:

RQ3: Which probabilistic prediction methods are most suitable for prediction of water demand?

Any of the above mentioned probabilistic methods is able to produce point forecasts too. Therefore, it would be sensible to compare the point forecasts from the probabilistic model with the point forecasts

of the deterministic methods.

RQ4: How do probabilistic models compare to deterministic models in terms of point prediction accuracy for prediction of water demand?

To evaluate if the models saved with optimal parameters achieve accurate predictions on unseen data demonstrating their generalization capability, the following research question is formulated:

RQ5: What is the performance difference between the validation phase and the test phase of the models?

In this context, generalization refers to the model's ability to maintain its predictive performance when applied to unseen data, such as the transition from validation to test data. For instance, a well-generalized model would achieve similar accuracy and error metrics on test data as observed during the validation phase, indicating its robustness in practical scenarios.

Lastly, it is important to examine when the prediction intervals of probabilistic models are too narrow or too wide, as this reflects under- or over-coverage. Under-coverage occurs when intervals fail to encompass the true values frequently enough, while over-coverage happens when intervals are unnecessarily broad, reducing their predictive performance and usability. Understanding these performances is essential for assessing the reliability of the models' uncertainty estimates across different conditions, such as variations throughout the year.

RQ6: When do the probabilistic models exhibit under/over-coverage?

Thesis Outline
Following this introduction, Chapter 2 reviews relevant literature, beginning with an overview of water demand characteristics 2.1, progressing to the next section where literature is reviewed of deterministic water demand literature 2.2. Next the background of probabilistic forecasts are explained in 2.3 followed with probabilistic forecasting literature in 2.4.

Chapter 3 outlines the materials and methods used in the study. It describes the dataset and preprocessing techniques in Section 3.1, followed by the metrics in Section 3.2 used to evaluate the models. Deterministic models along with their probabilistic extensions are elaborated in respectively Section and 3.3 and 3.4. Next the benchmark model is discussed in Section 3.5. The chapter also discusses the training methodology and hyperparameter optimization in Section 3.6. The chapter ends with the details of the design of the experiments in Section 3.7.

Chapter 4 presents the results, starting with the performance evaluations of the point forecasts in Section 4.1. Here the accuracy of point forecasts of the deterministic and probabilistic models are discussed as well as the generalization when moving from the validation to the unseen test data. In Section 4.2 the results of the probabilistic models are shown. It starts with the calibration results, as well as the coverages and sharpness on the test set. The section ends with the generalization when moving from the validation to the unseen test data of the interval forecasts.

Chapter 5 begins with providing a discussion of the methodologies and results in Section 5.1. In Section 5.2 recommendations are offered, giving insights in how to improve the methodologies as well as future research directions.

Finally, Chapter 6 concludes the thesis by answering the research questions.

Code base
The code developed for this thesis is publicly available to ensure reproducibility and to enable others to apply, adapt, or fork the project for their own research. The code is available at:
https://github.com/ChristiaanWewer/Thesis-Probabilistic-Water-Demand-Forecasting.

2. Literature Review

In this chapter the used literature is discussed. In Section 2.1 the process of water demand with its characteristics are discussed, followed by the literature review of deterministic forecasts methods in Section 2.2. Next the intricacies of probabilistic forecasts are described in Section 2.3. The chapter ends with the literature review of probabilistic water demand forecasts in 2.4.

2.1. Understanding Water Demand

Water demand is influenced by various factors beyond demographics and human behavior, including availability, climate, spatial properties, economic development, technological advancements in water-saving technologies, and the state of the water distribution system, which suffers losses due to aging. Other factors include water pricing and the population's commitment to environmental ethics [8, 52, 41, 37]. Demand fluctuations vary daily, weekly, monthly, and annually [35]. Residential water demand is primarily influenced by weather conditions such as temperature and precipitation, as well as social and economic activities. Linear relationships with weather patterns often break down, and weather effects may diminish as consumption nears a base level [58].

Individual consumption patterns are highly stochastic, whereas larger group consumption such as on a city or district level is more predictable [35]. Industrial water demand fluctuations are generally less stochastic. However, during the COVID-19 pandemic, which was happening at the start of 2021, the state of emergency and subsequent relaxations and restrictions influenced behavioral patterns and water demand. In general the pandemic decreased total water demand due to restrictions on economic sectors. Remote work policies shifted the spatial distribution of demand and altered peak usage times to [13].

Accurate insight in these influential patterns is difficult due to the complex relationships and not all of the above processes are measured or these data are difficult to access.

2.2. Deterministic Water Demand Forecast Literature

Several deterministic models have been succesfully applied in previous studies.
In [24] it was found a support vector regression model were slightly more accurate than RF (Random Forests), MLP and regression splines for districts in south-eastern Spain. Input features were temperature, wind velocity, atmospheric pressure and rain, whose predictive influences were not evaluated.
[5] compares a MLP, NARX (Nonlinear autoregressive exogenous model) and a SARIMA (Seasonal Autoregressive Integrated Moving Average) model with and without exogenous weather input. Of these models four variants are described, which forecast the hourly water demand value 1 hour ahead, 8 hour ahead, 12 hour ahead and 24 hour ahead. The case study of this research is the municipality district in Ontario, Canada. The model NARX with exogenous features shows slightly more accurate performance. Input features besides past water demand were hourly data of ambient temperature, dew point, absolute humidity, solar radiation, and station pressure, which were selected based on correlation.
The research [12] uses a SVR to forecast water demand for a district in Franca, Brazil. Rainfall, temperature, humidity, and wind velocity were chosen based on correlation with demand as features aside of past water demand.
The study [15] found that the addition of weather variables did not improve the forecasting performance. Furthermore LSTM, 1D-CNN (1D Convolutional Neural Network) and TCN (Temporal Convolutional Network) neural networks were compared. The models have competitive results with the TCN providing slighty more accurate results. Further with a global model trained on the districts together a more accurate model was made. Weather features were added but did not necessarily improve forecasts.
In the study [36] LSTMs are compared with SVR, RF and ARIMA in Heifei, China. The study shows LSTM models outperforming the other models. Additional input features such as weather variables had a limited to no influence. The temporal scale of the models were 15 minutes, 1 hour and daily.

When finalizing this research the papers for the Battle Of Water Demand Forecasting [6] were released, which uses the same dataset as this research. It was too short of notice to incorporate all submissions. The winning submission [32] proposes a method using an ensemble of 53 models, fitted separately on each DMA, including AR, GLM (generalized linear models), GAM (generalized additive models), and neural networks like MLP and LSTM. These individual models capture both linear (AR, GLM, GAM) and non-linear (MLP, LSTM) effects. The forecasts are combined using the smoothed Bernstein Online Aggregation (BOA) algorithm, which dynamically adjusts the weights of each model based on their past performance, allowing the ensemble to adapt to varying conditions and improve overall forecast accuracy. The second place approach [63] employs an ensemble of machine learning models, including variations of LightGBM, XGBoost, and WaveNet. The forecasting framework dynamically selects the top five best-performing models from a broader model pool using the Ensemble Reconciliation Strategy (ERS). The ERS evaluates model performance based on historical accuracy and combines their outputs by averaging to produce the final deterministic forecast. This ensemble method consistently outperformed individual models. Both submissions use a combination of weather, calendar and lagged water demand data.

2.3. Background Probabilistic Predictions

Probabilistic forecasting generates predictions that incorporate uncertainty by estimating a range of possible future values rather than a single point estimate. One approach to this is through predicting a full probability distribution, which provides detailed information about the likelihood of various outcomes. However, a more straightforward and interpretable method is interval-based forecasting, which outputs a prediction interval that specifies a range within which future values are expected to fall with a given probability.

Prediction intervals provide a straightforward way to quantify uncertainty. In this thesis a 95% prediction interval is used, which suggests that there is a 95% chance the true value will lie within the given bounds. While this method does not give precise probabilities for individual outcomes within the interval, it effectively captures the uncertainty by focusing on the overall range.

An important consideration in interval-based forecasting is heteroscedasticity, where the uncertainty (or variance) of future values is not constant and may vary depending on the input conditions. This can be due to external factors or fluctuations in the underlying data. Models that account for heteroscedasticity can adjust the width of the prediction intervals to reflect this changing uncertainty, providing more accurate forecasts. In contrast, models that assume homoscedasticity (constant variance) may not fully capture the variability in uncertainty, leading to less reliable intervals.

In theory there are two types of uncertainty that contribute to the prediction interval, namely epistemic and aleatoric uncertainty [26, 11]. Aleatoric uncertainty refers to the notion of randomness, for example a coin flip. In water demand context this refers to the uncertainty caused by unpredictable variations in consumer behavior. For example, individual water usage can vary widely due to factors like spontaneous activities or random consumption patterns, which cannot be precisely predicted or controlled. This data-generating process cannot be reduced by adding information. As opposed to this, epistemic uncertainty refers to the uncertainty caused by lack of knowledge and is in a deep learning context often understood as the uncertainty about the model parameters and input data. Epistemic uncertainty can be reduced with complete knowledge of the system, whereas aleatoric uncertainty remains irreducible. Within the system of water demand there is a lot of uncertainty. Water demand is mostly caused by direct human behavior in residential areas or by industrial processes in industrial areas. It is therefore difficult to know what exactly happens within each area. It is therefore impossible to completely remove the epistemic uncertainty as it will always be a substantial component.

It is difficult to know beforehand which probabilistic method will perform best, thus in this research four probabilistic methods are selected, of which three are often used. These are Monte-Carlo Dropout, Neural Quantile Regression, Gaussian Mixture Density Network and Conformalized Quantile Regression. The last method is a combination between two conformal prediction methods adapted for multi-step use, analyzing the effectiveness of conformal predictions. In terms of uncertainty Monte-Carlo Dropout gives a form of epistemic uncertainty namely model uncertainty [20, 11], where it captures the uncertainty associated with the model's parameters and structure. Neural Quantile Regression and

Gaussian Mixture Density Networks try to learn the uncertainty. Therefore these methods aim to capture and represent the inherent randomness within the data-generating process [11, 26, 50]. In terms of conformal predictions, the definitions of aleatoric and epistemic uncertainty remain unclear [26], but because conformal prediction methods construct prediction intervals from residuals, it is expected it captures indirectly both types of uncertainty.

2.4. Probabilistic Water Demand Forecasting Literature

The study [43] forecasts 24 hours water demand ahead for a resident complex in Qom Iran. Input variables used in this research were past water demand, temperature, cloud cover and wind-speed which were decomposed using wavelets. RF (Random Forests) did give more accurate predictions than SVR (Support Vector Machines), extreme learning machines and MLP. The prediction intervals were constructed by combining the variance of an ensemble of models and the variance of residuals of the training data.

In [19] a Markov Chain approach is considered with solely lagged water demand for probabilistic forecasts of the lead times 1, 2, 3, 6, 12, 18, 24 hour forward. The model is slightly more accurate than a MLP when it comes to point forecasts. The location is unclear the spatial scale is on district level.

The research [31] uses an ensemble of deterministic time series analysis models to forecast 24 hours ahead, using a separate model for each timestep. A GARCH (Generalized Auto Regressive Conditional heteroscedasticity) model was shown to give the most accurate results and an AR (Auto Regressive) model performs accurate too. Other models considered were RF, MLP, SVR and SARIMA. Calendar features were considered but climatological variables were not considered. The data used is from a water supplier in Western Germany. It is unclear on which spatial level this study is carried out.

Bayesian linear regression is used by [27] to generate probabilistic forecasts of water demand 24 hours in advance. The research only uses models with lagged values of water demand up to 168 hours back (1 week) and residuals of earlier forecasting steps. In [1] the Model Conditional Processor method is used to calibrate an ensemble of forecasts for the Castelfranco Emilia municipality in Italy. None of these papers analyzed how well their methods over the rest of the year or over the forecasting horizon. Additionally it was found that the probabilistic ensemble also improved point forecasts.

3. Materials and Methods

In this chapter the materials as well as the methods used to answer the sub-questions are explained. First the materials are described which consists of the dataset and processing of this in section 3.1. Subsequently the metrics in section 3.2 which are followed by the deterministic forecast models which are in section 3.3 explained. The chapter continues to explain the probabilistic extensions for the models in section 3.4 and the benchmark model in 3.5. Then the chapter explains the methods used to train the neural networks in 3.6. Lastly this chapter ends with Section 3.7 that describes the setup of the experiments.

3.1. Dataset and Data Processing

3.1.1. The Case Study In Ferrara, Italy

The dataset used for this project is water demand data of ten districts in the city of Ferrara, Italy [6]. The dataset consist of two years and two months of hourly data, which translates to 19056 rows. There are 10 columns where each represents an own district metered area (DMA).
Unfortunately, information about the location or the structure of the water distribution system is unavailable due to safety concerns. Having this knowledge would provide spatial insights, allowing for the creation of features and understanding whether the DMAs are hierarchically connected.

The data are measurements of water inflow to the districts of the city and do therefore include consumptions and leakages and vary in terms of characteristics per area.

Table 3.1: Water demand dataset Ferrara

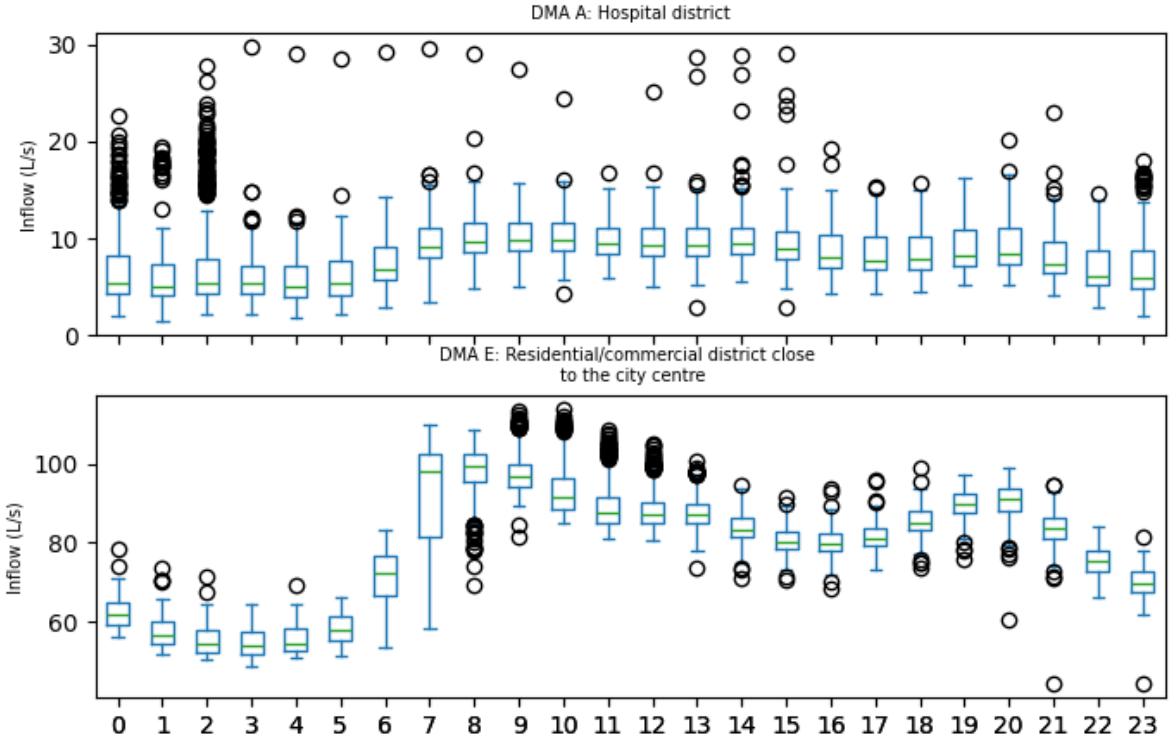
DMA ID	Area Characteristics	Numbers of Users Supplied	Average Inflow (L/s)
A	Hosptial district	162	8.4
B	Residential district in the countryside	531	9.6
C	Residential district in the countryside	607	4.3
D	Suburban residential/commercial district	2094	32.9
E	Residential/commercial district close to the city centre	7955	78.3
F	Suburban district including sport facilities and office buildings	1135	8.1
G	Residential district close to the city centre	3180	25.1
H	City centre district	2901	20.8
I	Commerical/industrial district close to the port	425	20.6
J	Commerical/industrial district close to the port	776	26.4

The water demand dataset has at most three distinct seasonailities, namely: annual, weekly and daily. As visible in figure 3.4 the hospital district has for example a very distinct seasonal pattern, as for example the residential district close to the city center has much less annual seasonality.

The demand patterns (appendix C) of DMA D and E have the largest means and with that also the largest deviations between the low and high inflows. The hospital district has many outliers from 00:00 to 2:00 throughout the week and not in the weekend. Most of the districts do have two peaks, one in the morning and one in the evening. This is barely the case in the hospital district, the office district and the industrial districts (DMA A, F, I and J) due to the behavioral changes between people and industry. In the residential districts the pattern of leveryday life is visible, people wake up in the morning, leave to work and have their peak in water use around 20:00-22:00. The outliers are the most discernible difference between weekdays in weekends, besides for both industrial districts (DMA I and J) where the water use is reduced. The outliers are otherwise primarily the differences between weekdays and weekends.

Figure 3.1: Location case study Ferrara

Figure 3.2: Boxplot water demand Hospital District and Residential Distrct per hour of the day.



Further DMA A, the hospital district, does have peaks which change pattern every while, which can be due to some device that requires plentiful water. This pattern is in the beginning of the dataset in the beginning of 2021 at Tuesdays, then in the beginning of 2022 it shifts to Tuesdays and Mondays, In the end of 2022 it shifts to Sundays and Tuesdays.

Figure 3.3: Steps Data Processing

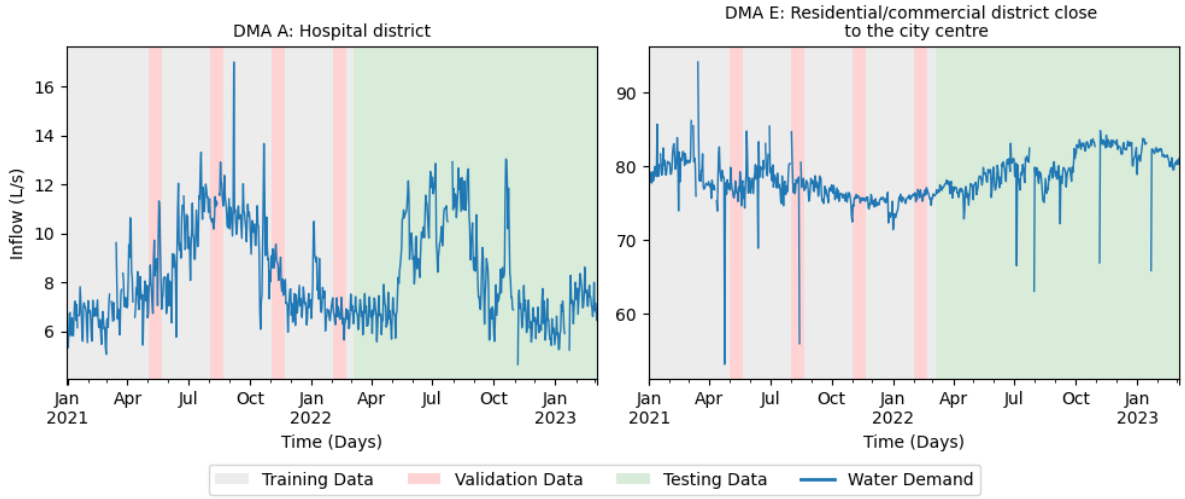
3.1.2. Data Processing

The data were processed with the following steps.

1. Data Splits And Missing Values Deep learning requires three data splits: training, validation, and testing. The training set fits the model, while the validation set, assessed during training, helps avoid over fitting by saving iterations with minimal error. Over fitting occurs when a model shows low error on the training set but high error on the validation set.

The dataset spans only two years and two months, presenting challenges for splitting due to annual seasonality. Ideally each split covers at least a year to encompass the full distribution of the largest season. The first year and two months serve for training and validation, with several smaller segments forming the validation set to have a similar distribution (see Appendix B.4 for figures for each DMA). The remainder of data are used for training. The entire final year acts as the testing split. This is visible in figure 3.4 below and in the appendix B.3 for all the DMAs.

Figure 3.4: Water demand of two districts with data splits. As visible the seasonality is still captured for the validation data however not every district has a strong annual seasonality



Before sequences were created as explained in the next paragraph, the data are normalized by subtracting the mean and dividing by the standard deviation. Each of the three datasets are normalized with the the mean and standard deviation of the training split.

2. Creating Sequences To use neural networks for this project, the data need to be shaped into the right format. These are 168 values of lagged water demand as an input sequence and 24 values of the consecutive water demand to compute the errors/losses. These sequences could not have any missing values. Given the significant number of missing values, the data from the first year and two months (training + validation data) were interpolated per segment, limited to three consecutive steps to reduce bias. This is limited to three steps because up to three steps there is a significant number

of sequences that could be constructed. Four, five or six steps had found to be of minimal benefit. Without interpolation it was not possible to make enough sequences per split. Some DMAs would have no validation data left. By enlarging the splits there would only be few training sequences for that DMA. This is due to the padding required for the datasets. If the splits would be made in the second year of data it would have been difficult to assess the generalization of the models because the distributions would be too similar between the validation and testing sets. By trial and error each of the validation splits were chosen to be 21 days long and start at (dd-mm-yyyy format) 01-05-2021, 01-08-2021, 01-11-2021, 02-02-2022. The training set consists of the data from from 01-01-2021 to 05-03-2022 minus the splits of the validation data. The testing set is from 05-03-2022 to 05-03-2023. This way the training splits consists totally of 44.5%, validation split of 8.5% and the testing split of 47.0%. The number of sequences per split are visible in Appendix B.2. An input sequence length of 168 was taken as a longer sequence length decreases the number of sequences that can be constructed as well. The largest auto-correlations do not have lags that are longer than this period as well B.4.1.

For the models that were trained on all the models the training sequences were combined between different DMAs and subsequently uniformly shuffled. The sequences of the validation data were kept separately.

3.2. Metrics

3.2.1. Deterministic Metrics

The metrics for point forecasts include the mean absolute error (MAE), median absolute scaled error (RMAE), and symmetric mean absolute percentage error (MAPE). Lower values indicate better performance for these metrics.

The mean absolute error quantifies the average magnitude of errors in a set of predictions

$$\text{MAE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |y_{i,j} - \hat{y}_{i,j}|, \tag{3.1}$$

where $\mathbf{y}$ and $\hat{\mathbf{y}}$ are ground truth and prediction matrices respectively, represented as NxM matrices where i denotes each separate forecast/ground truth and j each time step of this forecast/ground truth. N are the number of forecasts and M is the forecasting horizon.

The RMAE (relative mean absolute error) compares error those of a benchmark forecast by dividing the MAE of the forecast by the MAE of the benchmark forecast. This metric allows comparison between different DMAs and shows the usefulness of the developed forecasting model as an error larger than one means the benchmark forecast is more accurate when evaluated over the time series.

$$\text{RMAE}(\mathbf{y}, \hat{\mathbf{y}}, \mathbf{y}^*) = \frac{\text{MAE}(\mathbf{y}, \hat{\mathbf{y}})}{\text{MAE}(\mathbf{y}, \mathbf{y}^*)} \tag{3.2}$$

The MAPE metric, also scale-free, measures the relative error as a percentage of the true forecast

$$\text{MAPE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{100}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{|y_{i,j} - \hat{y}_{i,j}|}{|y_{i,j}|}. \tag{3.3}$$

The MAPE metric is chosen as it is scale-free and thus makes the study comparable between different DMAs and other studies and is intuitive to interpret as a percentage. For example if the true value is 10, and the forecast is ± 1 off, the MAPE will give a 10% error.

To assess model generalization, a metric similar to the RMAE is employed, the GR (Generalization Ratio). This involves dividing the MAE of the testing set by the MAE of the validation set. A ratio greater than one suggests poor generalization, indicating that the model has overfitted to the first year of data rather than capturing patterns that apply to the testing period. In contrast, a ratio close to or below one shows better generalization, as the model identifies the pattern that extends to unseen data

$$\text{GR}(\mathbf{y}_{\text{test}}, \mathbf{y}_{\text{val}}, \hat{\mathbf{y}}_{\text{val}}, \hat{\mathbf{y}}_{\text{test}}) = \frac{\text{MAE}(\mathbf{y}_{\text{test}}, \hat{\mathbf{y}}_{\text{test}})}{\text{MAE}(\mathbf{y}_{\text{val}}, \hat{\mathbf{y}}_{\text{val}})}, \tag{3.4}$$

where test and val denote the testing and validation set. Note that the validation set used is of different length than the training length and consist of different parts within the year thus caution is required with the interpretation of this metric. Due to the missing data between different districts this differs per DMA too.

3.2.2. Probabilistic Metrics

Probabilistic forecasts are evaluated using the Prediction Interval Coverage Probability (PICP), Coverage Gap (CG), Prediction Interval Normalized Average Width (PINAW) and the Winkler Score (WS) that combines both concepts. The criterion to save the best quantile model is the Relative Quantile Error (RQE) and explained too.

The PICP measures the probability that the real water demand is within the predicted interval and is often referred to as coverage

$$\text{PICP}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbb{1}(y_{i,j} \in \{\hat{y}_{i,j}^{(0.025)}, \hat{y}_{i,j}^{(0.975)}\}), \tag{3.5}$$

To analyze the coverage over the entire time horizon a rolling version of the PICP is used. Here the PICP is computed based on the previous weeks forecasts $\text{PICP}_t(\mathbf{y}_{t-168:t}, \hat{\mathbf{y}}_{t-168:t})$. Note that within the interval $t - 168$ to t the forecasted values that lie within there are used. Not the forecasts that are generated within there. The difference of coverage between weekends and weekdays are analyzed too, by subtracting the coverage of the weekend by the coverage of the weekday.

The CG (Coverage gap) measures how far the probabilistic forecasts are from the target prediction interval. A lower value indicates the prediction intervals are closer to the target prediction interval

$$\text{CG}(\mathbf{y}, \hat{\mathbf{y}}) = |1 - \text{PICP}(\mathbf{y}, \hat{\mathbf{y}})| \tag{3.6}$$

where $\alpha = 0.05$ and indicates the target 0.95 prediction interval.

The PINAW calculates the normalized average width of prediction intervals, referred to as sharpness, using the formula

$$\text{PINAW}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{NMR} \sum_{i=1}^{N} \sum_{j=1}^{M} (\hat{y}_{i,j}^{(0.975)} - \hat{y}_{i,j}^{(0.025)}), \tag{3.7}$$

where originally R represents the maximum range in the ground truth data. A lower value indicates a sharper interval. In this research the PINAW is modified to be more robust by using the IQR (Inter Quartile Range) for the range. This is the distance between the 0.75th and 0.25th quantile.

Finally the Winkler score [57] combines both the coverage and the width. The Winkler score is chosen because this metric is a proper scoring rule [21], meaning that following the metric the better forecast always has a better score. Other metrics that combine the coverage and the width of the forecasts that combine the PICP and PINAW more explicitly in a metric, such as the Coverage Width-Based Criterion (CWC) are not proper scoring rules [40] and requires tuning the width-coverage tradeoff, adding potential bias and ambiguity.

The Winkler Score measures the width of the prediction interval and if the true water demand is outside the prediction interval, the score is penalized with the distance this value is outside the interval weighted with the significance level of the prediction interval α

$$\text{WS}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left[\hat{y}_{i,j}^{(0.975)} - \hat{y}_{i,j}^{(0.025)} + \frac{2}{\alpha} \left(\hat{y}_{i,j}^{(0.025)} - y_{i,j} \right) \mathbb{1}\left(y_{i,j} < \hat{y}_{i,j}^{(0.025)} \right) + \frac{2}{\alpha} \left(y_{i,j} - \hat{y}_{i,j}^{(0.975)} \right) \mathbb{1}\left(y_{i,j} > \hat{y}_{i,j}^{(0.975)} \right) \right], \tag{3.8}$$

where $\alpha = 0.5$ indicating the 0.95 prediction interval. This score is not scale independent, thus in this project the score is divided by the Mean Winkler Score of the benchmark model, similar to the RMAE.

To find which model has best performance the Winkler Score is used. In case there is a low coverage (PICP) on the testing set with a high Winkler Score, we might disprefer the model. Thus we finally rank the models the following

$$\text{CWS} = \begin{cases} \text{WS} & \text{if PICP} > 0.95 - \eta \\ - & \text{otherwise} \end{cases} \tag{3.9}$$

where η is the maximum drop we allow relative to the the preferred 0.95 coverage probability. We call this score the Conditional Winkler Score (CWS). Because the QR and MDN models were difficult to calibrate a second version is created where the preferred coverage is not 0.95 but the PICP on the validation set, assuming it was

$$\begin{cases} \text{WS} & \text{if PICP} > \text{PICP}_{\text{val}} - \eta \\ - & \text{otherwise} \end{cases} \tag{3.10}$$

Because η is an arbitrary value the score is computed for values from 0.01 to 0.05 and then we count which models perform best per DMA.

The metric RQE (Relative Quantile Error) is used during the training of the probabilistic models, which is the quantile loss function of the model divided of the quantile loss function of the benchmark model

$$\text{RQE} = \frac{1}{10} \sum_{\text{DMA}} \frac{\mathcal{L}_{Q,\text{DMA}}(\mathbf{y}_{\text{val}}, \hat{\mathbf{y}}_{\text{val}})}{\mathcal{L}_{Q,\text{DMA}}(\mathbf{y}_{\text{val}}, \mathbf{y}^*_{\text{val}})}. \tag{3.11}$$

Note that even though this metric uses the quantile loss function, it does not actually update the losses. Instead it is used to select the best model using the validation data. The quantile loss function is defined and elaborated in 3.4.2.

To analyze the generalizability of the probabilistic models between the testing and validation set, the differences between the models are analyzed in terms of coverage and sharpness. This makes it is possible to understand how the model performance differs when navigating from the validation set to the unseen test set.

$$\Delta\text{PICP} = \text{PICP}_{\text{val}} - \text{PICP}_{\text{test}} \tag{3.12}$$

$$\Delta\text{PINAW} = \text{PINAW}_{\text{test}} - \text{PINAW}_{\text{val}} \tag{3.13}$$

where the test and val indicate the metrics on respectively the testing and validation set.

3.3. Deterministic Forecasting Models

3.3.1. Linear Model

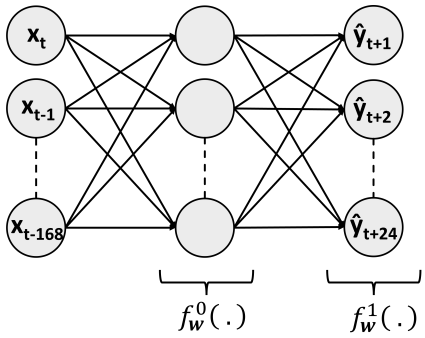
The Linear Model is a feed-forward neural network that uses two linear layers with dropout regularization. To maintain the network's linearity, a linear/identity activation functions are used.

A linear layer follows the equation:

$$f^{(l)}_{\mathbf{w}}(\mathbf{x}) = \mathbf{b}^{(l)} + \mathbf{w}^{(l)}\mathbf{x}, \tag{3.14}$$

where $\mathbf{w}$ are the learned parameters, $l \in \{1, .., L\}$ represents the number of layers with L being the maximum number of layers, and x is the input data. When $l > 1$, x is the output from the previous layer, $f^{(l-1)}_{\mathbf{x}}$. Note that neither the number of layers nor neurons should affect the model's predictive power (as long as they are equal or more than the minimum number of input and output data values), as they can be mathematically rewritten into a single linear layer.

Figure 3.5: Overview linear model



The output after the first linear layer is Dropout regularization [54, 49] is utilized to turn a neuron on or off with probability p, to increase the generalization of the model. To make sure the output values do not change magnitude the neurons that are not turned off are scaled with $\frac{1}{p}$.

3.3.2. Multi-layer Perceptron Model

The Multilayer Perceptron (MLP) is similar to the previously discussed linear model but uses activation functions between each linear layer.

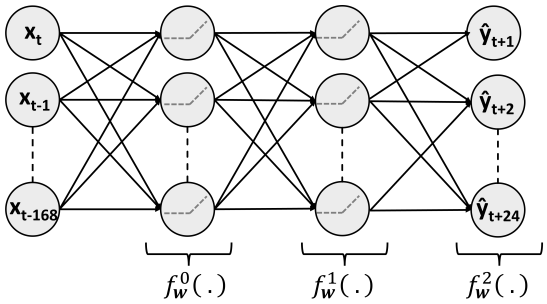
$$f_{\mathbf{w}}^{(l)}(\mathbf{x}) = \phi(\mathbf{b}^{(l)} + \mathbf{w}^{(l)}\mathbf{x}), \tag{3.15}$$

where $\mathbf{w}$ are the learnable parameters, $l \in \{1, .., L\}$ represents the number of layers with L being the maximum number of layers, and $\mathbf{x}$ is the input data. When $l > 1$, x is the output from the previous layer, $f_{\mathbf{w}}^{(l-1)}$. Additionally, ϕ is the ReLU activation function

$$\phi(\mathbf{x}) = \max(0, \mathbf{x}). \tag{3.16}$$

The Multilayer Perceptron includes an input layer that transforms the input data to a specific hidden size, which defines the number of neurons. Subsequently, several hidden layers are used, each with the same number of neurons. Finally, an output layer transforms the last hidden layer to the size of the output, without an activation function.

Figure 3.6: Overview of a Multi-Layer Perceptron model with two hidden layers



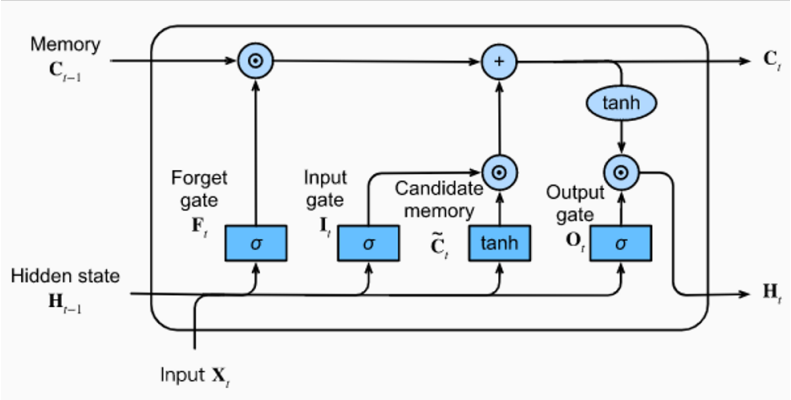
The hyperparameters are the number of hidden layers, the number of neurons in each hidden layer, the dropout rate, and the learning rate.

3.3.3. Long Short-Term Memory Model

A Long Short-Term Memory Model (LSTM) [25] is a type of recurrent neural network that is designed to learn long- and short-range dependencies in sequential data. The long-range dependencies are modelled by a 'conveyor belt' (cell state $\mathbf{C_t}$), over the input sequence.

At each step in the sequence information is added and removed by an input gate and a forget gate. The output gate computes using the cell state the next hidden state.

Figure 3.7: LSTM Cell



The first computation is the forget gate to learn which information should be removed from the cell state $\mathbf{C_t}$.

$$\mathbf{F_t} = \sigma(\mathbf{w}_{f1}\mathbf{x}_t + \mathbf{b}_{f1} + \mathbf{w}_{f2}\mathbf{H}_{t-1} + \mathbf{b}_{f2}), \tag{3.17}$$

where $\mathbf{w}$ and $\mathbf{b}$ are the weights and σ the sigmoid activation function

$$\sigma(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}}}, \tag{3.18}$$

where the range of this activation function is $(0, 1)$. Subsequently the input gate and candidate memory cell are computed

$$\mathbf{I_t} = \sigma(\mathbf{w}_{i1}\mathbf{x}_t + \mathbf{b}_{i1} + \mathbf{w}_{i2}\mathbf{H}_{t-1} + \mathbf{b}_{i2}), \tag{3.19}$$

$$\tilde{\mathbf{C}}_\mathbf{t} = \tanh(\mathbf{w}_{g1}\mathbf{x} + \mathbf{b}_{g1} + \mathbf{w}_{g2}\mathbf{H}_{t-1} + \mathbf{b}_{g2}), \tag{3.20}$$

These are combined by

$$\mathbf{C_t} = \mathbf{F_t} \odot \mathbf{C}_{t-1} + \mathbf{I_t} \odot \tilde{\mathbf{C}}_\mathbf{t}, \tag{3.21}$$

The candidate cell $\tilde{\mathbf{C}}_\mathbf{t}$ generates new information to add to the cell state, while the input gate $\mathbf{I_t}$ filters this information using element-wise computation ($\odot$), similar to the forget gate. This process weighs the information with the sigmoid activation function, where outputs between 0 and 1 determine the importance: 0 means the information can be forgotten, and 1 means it is important.

Lastly the output gate ($\mathbf{O_t}$) weighs similarly to the previous gates how much of the cell state is used to compute the hidden state.

$$\mathbf{O_t} = \sigma(\mathbf{w}_{o1}\mathbf{x} + \mathbf{b}_{o1} + \mathbf{w}_{o2}\mathbf{H}_{t-1} + \mathbf{b}_{o2}), \tag{3.22}$$

$$\mathbf{H_t} = \mathbf{O_t} \odot \tanh(\mathbf{C_t}). \tag{3.23}$$

The implemented LSTM concatenates the last hidden state H_t using a linear layer (see equation 3.14) to obtain the 24 output predictions.

The hyperparameters of the LSTM model include the number of layers, hidden size, dropout rate and learning rate.

One-Hot embedding LSTM

An one-hot embedding is used to introduce the categorical data to the LSTM, which is the DMA indicator in this research. These categorical data are transformed from a column with the DMA number to a standard basis vector

$$\text{For } k = 1, 2, \ldots, 10: \quad \text{DMA}_k \longrightarrow \mathbf{e}_k \tag{3.24}$$

Where the left side of the arrow are the DMAs and the right side are the one-hot encoded DMAs. The model will get the one-hot sequence and the lagged water demand data. Subsequently it will transform each index of the one-hot embedding with a linear layer to a predefined output size ending with the Sigmoid activation function to map the outcome to a probability between 0 and 1

$$\mathbf{X}_{e,t} = \sigma(\mathbf{w}\mathbf{e}_k), \tag{3.25}$$

where $\mathbf{X}_e$ is the final outcome of the one-hot encoding.

The concept of this one-hot embedding is the model can learn distinctive characteristics per DMA. The MLP does not require an embedding like this because it does not process the data as a sequence.

3.4. Probabilistic Extensions

3.4.1. Monte-Carlo Dropout

Monte-Carlo Dropout (MCD) [20] is an approximate Bayesian method for uncertainty quantification of deep learning models. The method extends dropout regularization of a deep neural network, where the dropout mechanism gives the neural network the ability to randomly turn each neuron off. This creates a different ensemble member within the neural network every time a forward pass is performed. By performing M forward passes through a trained model, an empirical distribution is obtained that represents the model uncertainty of the predictions

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}) \approx \{f_{\mathbf{w},\mathbf{d}_1}(\mathbf{x}), f_{\mathbf{w},\mathbf{d}_2}(\mathbf{x}), ..., f_{\mathbf{w},\mathbf{d}_M}(\mathbf{x})\}, \tag{3.26}$$

where $\mathbf{y}$ is the ground truth, $\mathbf{w}$ are the models weights, $\mathbf{x}$ the data passed through the model and f is the neural point prediction model. $\mathbf{d}_i$ is a dropout mask sampled from a Bernoulli distribution and T are the number of samples taken from this distribution. During each forward pass in the training process, nodes are randomly set to zero according to a Bernoulli distribution. The use of the Bernoulli distribution allows each node to independently have a probability p of being set to zero, which is known as the dropout rate. When a node is set to zero, it does not contribute to the forward pass. To compensate for the reduced number of active nodes, the remaining nodes are scaled by a factor of $\frac{1}{p}$, ensuring the overall magnitude of the activations remains consistent.

The prediction set $\hat{\mathbf{C}} = \{\hat{\mathbf{y}}^{(0.025)}, \hat{\mathbf{y}}^{(0.975)}\}$ for the 0.95 prediction interval were obtained by taking the empirical 0.025 and 0.975 quantiles of the empirical distribution in formula 3.26.

Calibration of the Monte-Carlo Dropout model requires finding a dropout rate that minimizes the coverage gap. Instead of the recommended grid search over dropout probabilities [20], a dichotomic search is employed. This uses the principle that larger dropout rates result in wider prediction intervals.

3.4.2. Neural Quantile Regression

Neural Quantile Regression is quantile regression with a neural network. The network directly estimates each quantile with the neural network and is trained by modifying the loss function with the mean quantile loss function [53, 56]

$$\mathcal{L}_Q(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{3T} \sum_{k=1}^{3} \sum_{t}^{T} \max[q_k \mathbf{e}^{(q)}_{t+1:t+T}, (q_k - 1)\mathbf{e}^{(q)}_{t+1:t+T}], \tag{3.27}$$

with the error $\mathbf{e}_k$, a vector with the size of the output sequence

$$\mathbf{e}^{(q)}_{t+1:t+T} = \mathbf{y}_{t+1:t+T} - \hat{\mathbf{y}}^{(q)}_{t+1:t+T}, \tag{3.28}$$

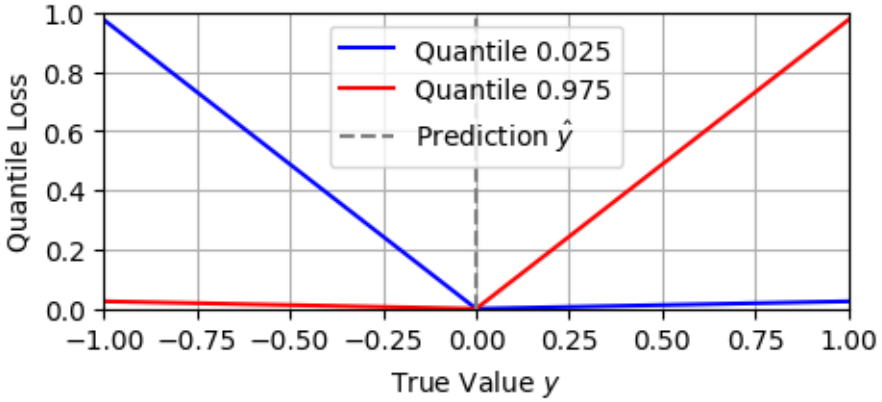
where $\mathcal{L}$ defines the loss function, $\hat{\mathbf{y}}^{(q)}_{t+1:t+T}$ and $\mathbf{y}_{t+:t+T}$ are respectively the quantile predictions and ground truth, q_k the quantile ($q_k \in \{0.025, 0.5, 0.975\}$). The max operator goes over the forecasting horizon length of $\mathbf{e}_{t+1:t+T}$. The quantile loss function assigns different weights to errors based on the quantile q_k. For quantiles where $q_k < 0.5$, the loss function penalizes overpredictions by a factor of q_k, meaning the error is multiplied by q_k for positive residuals (overpredictions), while underpredictions are

Algorithm 1 Dichotomic Search for Monte Carlo Dropout Rate

Input:
  Validation Dataset: $\mathbf{x}_v$, $\mathbf{y}_v$
  Number of samples: $n_{\text{samples}} = 1000$
  Quantiles: quantiles
  Desired coverage: PI $= 0.95$
  Maximum coverage gap: max_cov_gap $= 0.005$
  Maximum iterations: max_iterations $= 5$
  Initial bounds for dropout rate: $r_{\text{low}} = 0.05$, $r_{\text{high}} = 0.95$
Output: Optimal dropout rate r

Procedure:
Initialize $c \leftarrow 0$
Set coverage_gap $\leftarrow |\text{PICP} - \text{PI}|$
while coverage_gap $>$ max_cov_gap **do**
  if $c >$ max_iterations **then**
    break
  end if
  Compute new $r \leftarrow \frac{r_{\text{low}} + r_{\text{high}}}{2}$
  Compute prediction interval $\hat{\mathbf{y}}_v \leftarrow f_{\text{mcd}}(\mathbf{x}_v, r)$
  Compute PICP$(\mathbf{y}_v, \hat{\mathbf{y}}_v^{(0.025, 0.975)})$
  Update coverage_gap $\leftarrow |\text{PICP} - \text{PI}|$
  if PICP $<$ PI **then**
    Update $r_{\text{low}} \leftarrow r$
  else
    Update $r_{\text{high}} \leftarrow r$
  end if
  Increment $c \leftarrow c + 1$
end while
Return r

Figure 3.8: Quantile loss function



penalized by a factor of $(q_k - 1)$. This imbalance in penalties drives the model to favor lower predictions to minimize the higher cost associated with overpredictions. For instance, at $q_k = 0.025$, the model is expected to overpredict 2.5% of the time and underpredict 97.5% of the time, aligning with the definition of quantiles. Thus by estimating the 0.025 and 0.975 quantiles via this loss the 95% prediction interval is obtained. Note that for the median (0.5th quantile) the loss is proportional to the mean absolute error.

3.4.3. Gaussian Mixture Density Network

The Gaussian Mixture Density Network estimates the full distribution of future water demand by parameterizing it with a fixed number of Gaussian distributions. The output layer of the neural network is modified to predict for each time-step a weight, mean and standard deviation, resulting in the parameters to create the Gaussian mixture

$$\left\{ p(\hat{y}_i \mid \mathbf{x}_{t-L:t}) = \sum_{j=1}^{G} \pi_{i,j} \, \mathcal{N}(\hat{y}_i \mid \mu_{i,j}, \sigma_{i,j}) \right\}_{i=t+1}^{t+T} \tag{3.29}$$

where the set denotes each prediction step towards the horizon of the prediction and the sum denotes the mixture of Gaussians. The weights for each mixture are denoted by $\pi_{i,j}$ and the mean and standard deviation of each Gaussian are denoted by $\mu_{i,j}$ and $\sigma_{i,j}$, which are parameterized by the network. This parameterization is not directly done by the network, but those values require transformation

$$\pi_{i,j} = \frac{\exp\left(\hat{y}_{\pi_{i,j}}\right)}{\sum_{k=1}^{G} \exp\left(\hat{y}_{\pi_{i,k}}\right)}, \; \mu_{i,j} = \hat{y}_{\mu_{i,j}}, \; \sigma_{i,j} = \mathsf{ELU}(\hat{y}_{\sigma_{i,j}}) + 1.1, \tag{3.30}$$

where $\hat{y}_{\pi_{i,j}}$, $\hat{y}_{\mu_{i,j}}$ and $\hat{y}_{\sigma_{i,j}}$ are the direct outputs of the neural network that are transformed to the inputs of the mixture density model. The softmax distribution is used to make sure the weights ($\pi_{i,j}$ sum up to one for each separate mixture as the original paper suggests [9]. By having the sum of all weights to be 1 the output distribution is maintained. The means are raw outputs of the neural network and thus not transformed. The original paper suggests using the exponent function to have positive values for the standard deviation but since this makes the outcome very sensitive to larger values an ELU function is used as proposed in [11].

$$\mathsf{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0, \end{cases} \tag{3.31}$$

where $\alpha = 1$. The ELU is a linear function on the range $(0, \infty)$ and exponential on the range of (-1,0]. To have a positive value and to avoid the Gaussian to collapse, a value of 1.1 is added to the ELU. To avoid the Gaussian to collapse or be redundant, the standard deviation is clipped by only allowing values between 0.1 and 10.

he employed loss function is the negative log-likelihood, as used in [11].

$$\begin{aligned}\mathcal{L}_G[\mathbf{y}, p(\hat{\mathbf{y}}|\mathbf{x})] &= -\ln\left[p\left(\hat{\mathbf{y}}_{t+1:t+T}|\mathbf{x}_{t-L:t}\right)\right] \\ &= -\sum_{i=t}^{t+T} \mathbf{1}_{t+1:t+T} \ln\left\{\sum_{j=1}^{G} \pi_{i,j}\mathcal{N}\left(y_i \mid \mu_{i,j}, \sigma_{i,j}\right)\right\} \\ &= -\sum_{i=t}^{t+T} \mathbf{1}_{t+1:t+T} \left\{\ln \sum_{j=1}^{G} \exp\left[\ln \mathcal{N}\left(y_i \mid \mu_{i,j}, \sigma_{i,j}\right) - \ln \pi_{i,j}\right]\right\}\end{aligned} \tag{3.32}$$

To acquire the predictions 1000 samples were taken from each mixture of which the 0.025 and 0.975 quantiles were computed, obtaining the 95% prediction interval.

The method gives an extra hyperparameter which are the number of Gaussians used per mixture. Further there needs to be taken notice that the method increases the number of parameters of the model because each Gaussian added to the mixture multiplies the parameters of the last layer with $3 \cdot 24 = 72$.

3.4.4. Conformalized Quantile Regression

Conformalized Quantile Regression (CQR) adds an additional calibration step to the standard quantile regression framework. Specifically, conformal prediction uses the residuals (errors) from a portion of the data to recalibrate the predicted quantiles, ensuring that the prediction intervals are more reliable and better aligned with the data distribution.

However, in this research, the original method for CQR proposed by Romano et al. [45] is not directly applied. The main reason is that the original approach tends to produce prediction intervals that are too wide for time series data, as noted by Jensen et al. [29]. To address this issue, a modified version similar to that of Jensen et al. [29] is used. Unlike the method in that work, here the correction for finite sample sizes is incorporated, and the median quantile predictions are also conformalized. The first change is minimal and aims to improve the prediction interval sizes due to the smaller dataset size. Ideally the dataset size is supposed to have 1000 data points [2, 48], but this was not always possible as shown in B.2. The second adaptation aims at adjusting the median prediction to be in line of the median of all past predictions.

To perform the recalibration, the residuals (or nonconformity scores in conformal prediction terminology) for each quantile are computed using a validation data split:

$$\epsilon_v^{(q)} = \mathbf{y} - \hat{\mathbf{y}}_v^{(q)}, \tag{3.33}$$

where $\epsilon_v^{(q)}$ represents the residuals for quantile q (typically $q \in 0.025, 0.5, 0.975$), $\mathbf{y}$ is the true value, and $\hat{\mathbf{y}}_v^{(q)}$ is the predicted quantile on the validation set. These residuals are then used to recalibrate the quantile predictions, with a correction for finite sampling, as follows:

$$\hat{\mathbf{y}}_{\text{calibrated}}^{(q)} = \hat{\mathbf{y}}^{(q)} + \text{Quantile}_q(\epsilon_v^{(q)}). \tag{3.34}$$

This recalibration process relies on the assumption of exchangeability between the calibration data (validation set) and the data being forecasted. In time series, where this assumption can be problematic due to temporal dependencies, the residuals are updated every T forecasts (where T is the forecast horizon). This procedure, following the approach of Xu et al. [59] and Jensen et al. [29], ensures that future information does not influence the residual updates.

The only remaining assumption is that the residuals should be stationary and strongly mixing, meaning that the errors have no discernible patterns and are effectively noise. This ensures that the forecasting model captures the underlying process accurately, making the assumption of exchangeability unnecessary.

The algorithm implemented is as follows

Algorithm 2 Conformal Quantile Regression Forecasting with Residual Updating

Input:
  Trained Quantile Regression model: $f(x)$
  Validation set: $\mathbf{x}_v$
  Testing set: $\mathbf{x}_t$
  Quantiles: $[0.025, 0.5, 0.975]$
Output: Forecasts with updated residuals

Step 1: Compute Residuals for Each Quantile
Initialize an empty dictionary to store residuals for each quantile: residuals $= \{\}$
for each quantile q in $[0.025, 0.5, 0.975]$ **do**
  Compute the forecast for each point in $\mathbf{x}_v$ using $f(x)$ at quantile q:
    $\hat{\mathbf{y}}_v^{(q)} = f(\mathbf{x}_v, q)$
  Calculate residuals for each forecasted quantile in the validation set:
    $\epsilon_v^{(q)} = \mathbf{y} - \hat{\mathbf{y}}_v^{(q)}$
    $\epsilon_v^{(q)}$ will be an $N \times M$ matrix, where N is the number of forecasts and M is the horizon of the model
  Store the residuals for quantile q in the residuals dictionary
end for

Step 2: Initialize Queue for Forecast Storage
Initialize an empty queue: forecast_queue $= [\,]$

Step 3: Iterative Forecasting and Residual Updating
for for each input sequence $\mathbf{x}_{t-L:t}$ in $\mathbf{x}$ **do**
  Run the forecasting model $f(x)$ to generate forecasts for each quantile q in $[0.025, 0.5, 0.975]$
  Store forecasts in forecast_queue with their corresponding datetime indices
  for each forecast in forecast_queue **do**
    if the current datetime surpasses the datetime of the forecast in the queue **then**
      Update the residuals for the corresponding quantile q
      Remove the outdated forecast from the queue
      Recalibrate each quantile q of the forecast after updating the queue with:
        $\hat{\mathbf{y}}_{\text{calibrated}}^{(q)} = \hat{\mathbf{y}}^{(q)} + \text{Quantile}_q(\epsilon_v^{(q)})$
    end if
  end for
end for

Different to the previously cited CQR-algorithms this algorithm is adapted to make sure it can deal with gaps by using the forecast queue in the testing set as without those adaptions the time-lag between the residuals and the forecasts become more than just the forecasting horizon. The algorithm is ran separately per DMA.

3.5. Benchmark Model

The benchmark model predicts water demand by averaging up to four past values of water demand that occur at the same time of day, but from one to four weeks earlier. This approach accounts for the daily and weekly seasonalities in the demand patterns. n is a parameter that needs to be calibrated. For example, if $n = 4$, the model may fall behind in capturing rapid or significant shifts in demand patterns. The model is defined as

$$y_{t+1}^* = \frac{1}{n} \sum_{i=1}^{n} y_{t+1-168*i}, \tag{3.35}$$

where the star indicates the forecast is performed by the benchmark model. A limitation of this approach is the assumption of the weekly seasonality. These may vary between DMA or temporally for example between winter and summer. The model may also lag behind fast increases in the water consumption pattern. Further this model assumes the future water demand is always similar to a value of an aggregation of previous lags, which is not always true.

The probabilistic extension of the benchmark model computes the prediction intervals by projecting prediction intervals of a year before on the forecasts. The residuals are binned per month because of the annual seasonality

$$\{\epsilon_1, \epsilon_2, ..., \epsilon_{12}\} = \epsilon = \mathbf{y} - \mathbf{y}^*, \tag{3.36}$$

finally the probabilistic benchmark predictions

$$y_{t+1}^{*,(0.025)} = y_{t+1} + \text{Quantile}_{0.025}(\epsilon_i), \quad y_{t+1}^{*,(0.975)} = y_{t+1} + \text{Quantile}_{0.975}(\epsilon_i), \quad \text{with } y_t \in \epsilon_\mathbf{i}, \tag{3.37}$$

where t is the hour of the forecast and i is the month t belongs to. This ensures the PICP is on the validation series 0.95. Formally this approach of binning residuals and computing prediction intervals from them is a form of Mondrian Conformal Prediction [10]. The limitation of this approach is that it assumes that each monthly distribution of the first year of the data are exchangeable to each monthly distribution of the second year.

3.6. Training Method of Neural Networks

3.6.1. Updating the Parameters

The Adam (Adaptive Moment Estimation) [30] optimizer was used to update the parameters of all the models. The Adam optimizer is an extension of SGD [44] (Stochastic Gradient Descent).

A forward pass is performed where the training data is passed through the network in batches, to compute the output predictions. The loss is computed using these output predictions and the ground truth. Of this loss gradients are computed, which are back-propagated using the chain-rule through the entire neural network. SGD uses the following formula

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}, \tag{3.38}$$

where θ represents the parameter vector, η is the learning rate and $\mathcal{L}$ the output of the loss function. Adam improves upon this by computing adaptive estimates of lower-order moments. The first moment ($\mathbf{m}$) is an estimate of the average gradient which helps to smooth out noisy updates. The second moment ($\mathbf{v}$) normalizes the gradient with an online moving average to make sure the update step does not explode for large gradients.

The update rules for Adam are given by

$$\begin{aligned}
\mathbf{m}_t &= \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\nabla_\theta L_t, \\
\mathbf{v}_t &= \beta_2 \mathbf{v}_{t-1} + (1-\beta_2)(\nabla_\theta L_t)^2, \\
\hat{\mathbf{m}}_t &= \frac{\mathbf{m}_t}{1-\beta_1^t}, \\
\hat{\mathbf{v}}_t &= \frac{\mathbf{v}_t}{1-\beta_2^t}, \\
\theta &\leftarrow \theta - \eta \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon},
\end{aligned} \tag{3.39}$$

where $\mathbf{m}_t$ and $\mathbf{v}_t$ are the first and second moment estimates, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are exponential decay rates for the moment estimates, η is the learning rate, and $\epsilon = 1 \times 10^{-8}$ is a small constant to prevent division by zero. The values given here are the standard hyperparameters used in PyTorch and used for this research.

3.6.2. Training Loop And Bayesian Optimization

Training Loop After all batches of training data have been processed by the network, and the parameters have been updated, an epoch is considered complete. Following each epoch, the model is evaluated on the validation data, and performance metrics are used to assess its accuracy. If the new model performance surpasses the previous results, the model is saved. For models trained on combined district data, the average performance score across all DMAs was used.

This approach ensures that the best model is retained. To ensure the model outperforms the benchmark model uniformly across all districts, the RMAE was applied for deterministic models, while the RQE was used for probabilistic models, as explained in Section 3.2. This step-by-step approach is described in Algorithm 3.

Algorithm 3 Training Process with RMAE or RQE

Input:
  Number of epochs: 250
  Batch size: 256
  Training data: x_{train}
  Validation data: x_{val}
  Score functions: RMAE or RQE

Procedure:
Randomly shuffle x_{train}
Divide x_{train} into batches
for each epoch $i = 1$ to 250 **do**
  for each batch x_{batch} in x_{train} **do**
    Run model $f(x)$ on x_{batch}
    Compute loss function $\mathcal{L}$
    Update model parameters using $\mathcal{L}$
  end for
  Run model $f(x)$ on x_{val}
  if the model is trained on all DMAs together **then**
    for each DMA $d = A$ to J **do**
      Compute score function on validation data for DMA d
    end for
    Compute the average score function across all DMAs
  else
    Compute the score function on x_{val}
  end if
  if the average score function (for combined DMAs) or the single score function (for individual models) is better than the previous score **then**
    Save model parameters
  end if
end for

Bayesian Hyperparameter Selection To optimize the hyperparameters of the models, this research uses Bayesian optimization, a method that chooses new hyperparameters based on previous results. A GP (Gaussian process) is used as a surrogate model to predict how changes in the hyperparameters will affect model performance. This allows the algorithm to focus on the most promising hyperparameter settings and avoid testing every possible combination.

Instead of searching randomly, the algorithm uses a function to decide whether to explore new settings or improve on known ones. This approach reduces the number of tests needed, making the process more efficient.

By using Bayesian optimization with a Gaussian process, the models hyperparameters can be tuned effectively, finding the best settings faster than other methods like random or grid search.

3.7. Design of Experiments

Experiments of Deterministic Models The first research experiment is to find whether training a single model per district or a global model on all the districts is has a better performance. To find the hyper-parameter set with the best parameters a Bayesian hyper-parameter search was executed at for the three models, per DMA and with the DMAs together and with the DMAs together with a DMA-indicator. The hyperparameter ranges were for the individual models were selected as follows.

Table 3.2: Hyperparameter Configurations for Linear, MLP, and LSTM Models trained per DMA, each with 100 runs

Hyper Parameters	Linear Model	MLP	LSTM
Dropout Rate	0.1, 0.15, 0.2	0.1, 0.15, 0.2	0.1, 0.15, 0.2
Hidden Size	32, 64, 128	32, 64, 128	32, 64, 128
Hidden Layers	-	0, 1, 2, 3	1, 2
Learning Rate	1e-5 – 1e-2	1e-5 – 1e-2	1e-5 – 1e-2
Number Epochs	250	250	250
Batch Size	256	256	256
Number of Parameters	12,376 - 24,728	6,200 - 74,264	5,272 - 70,168

The 'Hidden Size' refers to the number of neurons in each layer. The global LSTM with DMA-indicator was modified to learn an embedding using a one-hot encoding (see subsection 3.3.3). For the linear model the DMA-indicator was not considered, as it only would be able to change the intercept of the linear formula the model is equal to. It would not be able to change the interaction of the model with the lagged water demand input. To make sure the effect of dropout is similar between the linear model and the other models they were given a similar number of neurons for the layer. The models trained on all DMAs were given the options in the hyper-parameter search to have more parameters, because training on 10 different districts is a more complex task.

Table 3.3: Hyperparameter Configurations for Linear, MLP, and LSTM Models trained on all DMAs, with and without DMA-indicator, each optimized with 100 runs

Hyper Parameters	Linear Model	MLP	LSTM	MLP (DMA Indicator)	LSTM (DMA Indicator)
Dropout Rate	0.1, 0.15, 0.2	0.1, 0.15, 0.2	0.1, 0.15, 0.2	0.1, 0.15, 0.2	0.1, 0.15, 0.2
Hidden Size	64, 128, 256	64, 128, 256	32, 64, 128	64, 128, 256	32, 64, 128
Hidden Layers	-	0, 1, 2	1, 2	0, 1, 2	1, 2
Learning Rate	1e-5 – 1e-2	1e-5 – 1e-2	1e-5 – 1e-2	1e-5 – 1e-2	1e-5 – 1e-2
Number Epochs	250	250	250	250	250
Batch Size	256	256	256	256	256
Number of Parameters	12,376 - 49,432	12,376 - 181,016	5,272 - 202,264	13,016 - 183,576	19,018 - 204,874
DMA Embedding Output Size	-	-	-	-	2 - 7

The loss function was the MAE (also named L1-loss) because this loss is equivalent to learning the median, which makes the point forecasts comparable with the probabilistic forecasts. The results of the hyperparameter optimizations are visible in Appendix D.1.1. The point forecasts were compared with the MAPE and RMAE metrics, as averages and over the forecasting horizon. The generalization between the validation and testing data is analyzed using the GS. Note that the different differences of the splits of data and the short duration of them require caution when analyzing this.

Experiments of Probabilistic Models To continue to find the best performing probabilistic model, the best performing deterministic model was used. This was the MLP trained on all DMAs without the DMA indicator, as the results show in Section 3.3. This model was extended with the probabilistic extensions described.

For the QR and the MDN the hyperparameter optimization was conducted with the following configuration ranges

Table 3.4: Hyperparameter Configurations for MDN and QR Models, trained on all DMAs, optimized with 100 runs of the bayesian hyperparameter optimization.

Hyper Parameters	MDN	QR Model
Dropout Rate	0.1, 0.15, 0.2	0.1, 0.15, 0.2
Hidden Size	64, 128, 256	64, 128, 256
Hidden Layers	0, 1	0, 1, 2
Number of Gaussians	1, 2, 3, 4	-
Learning Rate	1e-5 – 1e-2	1e-5 – 1e-2
Number Epochs	250	250
Batch Size	256	256
Number of Parameters	15,496 - 183,072	15,496 - 193,352

The MCD and CQR models required data to calibrate. Because of the short dataset size these models were calibrated as explained in their respective sections (3.4.1 and 3.4.4) on the validation data.

First the medians of the probabilistic models were compared with the deterministic model results using the MAPE and RMAE metrics and the generalization between the validation and testing sets was determined with the GR metric, similar to the results of the deterministic models.

The probabilistic forecasts are analyzed in terms of their coverage and by comparing coverage against sharpness over the forecast horizon for a set of DMAs. This demonstrates how these models balance these aspects. Ideally the coverage and sharpness remain similar over the forecasting horizon and the coverage is as close to 0.95 as possible, which is the aimed prediction interval. The sharpness is ideally as low as possible. If these coverage and sharpness do vary, a forecast is desired where the coverage remains as close to 0.95 as possible with a varying sharpness. A condition of this is that the intervals should not become so wide they become unusable and lose practical utility.

Lastly the probabilistic models are analyzed using the CWS where the coverage declines with 0.02 as an example. This decline is relative to 0.95 and relative to the PICP on the validation set. It was chosen to do this because the QR and MDN were difficult to calibrate as the results show later in this report in Section 4.2.1. For practical applications different conditional thresholds for the drop in coverage can be set that suit the goal. To give an overview, the DMAs are counted where each model performs best with declines of of 0.01 to 0.05.

For generalization the differences in terms of coverage as well as in terms of sharpness between the validation and testing set were analyzed. It is important to notice as mentioned before that these datasets have different lengths due to the way they are constructed as explained in 3.1.

4. Results

In Section 4.1 the results of the point forecasts are described, acquired from the deterministic models in Subsection 4.1.1. The results of point forecasts acquired from the probabilistic models are shown in Subsection 4.1.2 and the generalization of forecasts from both deterministic and probabilistic models in Subsection 4.1.3. In Section 4.2 the results of the prediction intervals of the probabilistic models are described which starts with the Subsection 4.2.1 which shows how well the probabilistic models are calibrated. Subsection 4.2.2 describes the results of the probabilistic forecasts in terms of coverage and sharpness. The generalization of these models is shown in 4.1.3. In the tables green selections of values are the best per specific DMA and the red selections of values are the worst according to the used metric. The average displayed in these tables is an average over all the DMAs.

4.1. Results Point Forecasts

4.1.1. Results Deterministic Models

The MLP model performs more accurate (see 4.1), when trained on all DMAs jointly than when trained per DMA. Here the DMA indicator has no significant influence on the results according to the RMAE or MAPE metrics.

Table 4.1: RMAE of MLP Models on Testing Set

	A	B	C	D	E	F	G	H	I	J	Average
MLP trained on all DMAs	0.79	**0.70**	**0.71**	**0.87**	**0.89**	0.91	**0.93**	0.96	**0.88**	0.87	**0.85**
MLP DMA Indicator trained on all DMAs	**0.80**	0.73	0.74	0.88	0.91	**0.90**	1.00	**0.95**	0.89	**0.86**	0.87
MLP trained per DMA	**0.78**	**0.87**	**0.85**	**0.94**	**1.36**	**0.92**	**2.05**	**1.48**	**0.96**	**0.92**	**1.11**

The results of the LSTM model showcase varying results with respect to the impact of the DMA indicator(see Table 4.2). On average the LSTM is more accurate when trained on all DMAs together. The model is on average slightly worse than the Benchmark Model because the RMAE is larger than 1. The LSTM with DMA indicator trained on all DMAs performs best on three DMAs, trained per DMA on two DMAs and on five DMAs it is best when trained on all the DMAs together without the DMA indicator according to the RMAE. According to the MAPE metric the LSTM trained on all the DMAs is more accurate on six DMAs and with DMA indicator also on three DMAs.

Table 4.2: RMAE of LSTM Models on Testing Set

	A	B	C	D	E	F	G	H	I	J	Average
LSTM trained on all DMAs	**0.87**	**0.81**	**0.84**	**0.93**	1.13	**0.99**	**1.39**	**1.18**	1.11	1.03	**1.03**
LSTM DMA Indicator trained on all DMAs	**0.84**	0.83	0.92	0.96	**1.05**	**0.99**	1.62	1.22	**1.13**	**0.96**	1.05
LSTM trained per DMA	0.86	**0.96**	**1.07**	**1.09**	**1.27**	**0.98**	**2.57**	**1.63**	**1.02**	**1.13**	**1.26**

The linear models perform very similar with no clear better performing model when trained per DMA or together (see Table 4.3). The performance is so similar the MAPE and RMAE metrics give conflicting results. For DMA B, D, E, F, G, H, I the linear model trained on all DMAs is slightly better. For the other DMAs the linear model trained per DMA is better. There is no DMA where one of the linear models performs substantially better than another model.

Table 4.3: RMAE of Linear Models on Testing Set

	A	B	C	D	E	F	G	H	I	J	Average
Linear Model trained on all DMAs	**0.84**	**0.72**	**0.73**	**0.92**	**1.10**	**0.92**	**0.89**	**0.98**	**0.87**	**0.98**	**0.90**
Linear Model trained per DMA	**0.80**	**0.72**	**0.72**	**0.94**	**1.12**	**0.93**	**0.98**	**1.03**	**0.87**	**0.96**	**0.91**

In Table 4.4 a subset of the results of all the models are visible, which are the best models of the previous three tables to maintain overview including the benchmark model. The full Table 4.6 is in Appendix F.1. The MLP, LSTM and Linear models trained on all DMAs and the Linear model trained per DMA as well as the benchmark model. In Appendix F.1 the full table is visible. All the models have difficulty in forecasting the hospital district (DMA A), due to the highest overall MAPE, as compared to the other districts. To a lesser extent this applies to the second residential district in the countryside (DMA C) and the suburban district with sport facilities (DMA F). The residential/commercial district close to the city center (DMA E) has the smallest MAPE error. One remarkable result is that the models perform a lot more accurate on DMA B compared to DMA C because both time series are similar in terms of their description, distributions as well as patterns (see Appendix B.4) and B.3). The benchmark model results has a similar order of magnitude as any of the neural network models.

Table 4.4: MAPE [%] Point Forecasts of deterministic models on test set

	A	B	C	D	E	F	G	H	I	J	Average
MLP trained on all DMAs	**13.52**	**4.71**	9.34	**6.96**	**1.81**	**8.86**	3.76	**4.52**	6.11	**4.67**	**6.43**
Linear Model trained per DMA	13.98	4.87	**9.30**	7.54	**2.33**	8.96	3.91	4.95	**6.08**	5.20	6.71
Linear Model trained on all DMAs	15.15	4.81	9.37	7.30	2.25	8.92	**3.53**	4.69	6.13	5.26	6.74
LSTM trained on all DMAs	14.88	5.35	11.31	7.37	2.32	9.53	**5.57**	**5.57**	**7.62**	5.50	7.50
Benchmark Model	**18.49**	**6.65**	**12.64**	**8.02**	2.06	**9.71**	4.00	4.80	7.08	**5.54**	**7.90**

The most accurate result are delivered by the MLP model which has an average MAPE of 6.43 and RMAE of 0.85. The MLP with DMA indicator performs slightly worse after that with an average MAPE of 6.71 and RMAE of 0.90, emphasizing that the addition of an indicator did not yield an improvement in forecasts The LSTM trained per DMA is the worst performing model (see Appendix F.1).

The Linear models perform very similar to the MLP trained on all DMAs with a MAPE of 6.71 trained per DMA and 6.74 trained on all DMAs or jointly.

Table 4.5: RMAE [-] Point forecasts of deterministic models on test set

	A	B	C	D	E	F	G	H	I	J	Average
MLP trained on all DMAs	**0.79**	**0.70**	**0.71**	**0.87**	**0.89**	**0.91**	0.93	**0.96**	0.88	**0.87**	**0.85**
Linear Model trained on all DMAs	0.84	0.72	0.73	0.92	1.10	0.92	**0.89**	0.98	**0.87**	0.98	0.90
Linear Model trained per DMA	0.80	0.72	0.72	0.94	1.12	0.93	0.98	1.03	**0.87**	0.96	0.91
LSTM trained on all DMAs	**0.87**	0.81	0.84	0.93	**1.13**	**0.99**	1.39	1.18	1.11	**1.03**	1.03
LSTM DMA Indicator trained on all DMAs	0.84	**0.83**	**0.92**	**0.96**	1.05	**0.99**	**1.62**	**1.22**	**1.13**	0.96	**1.05**

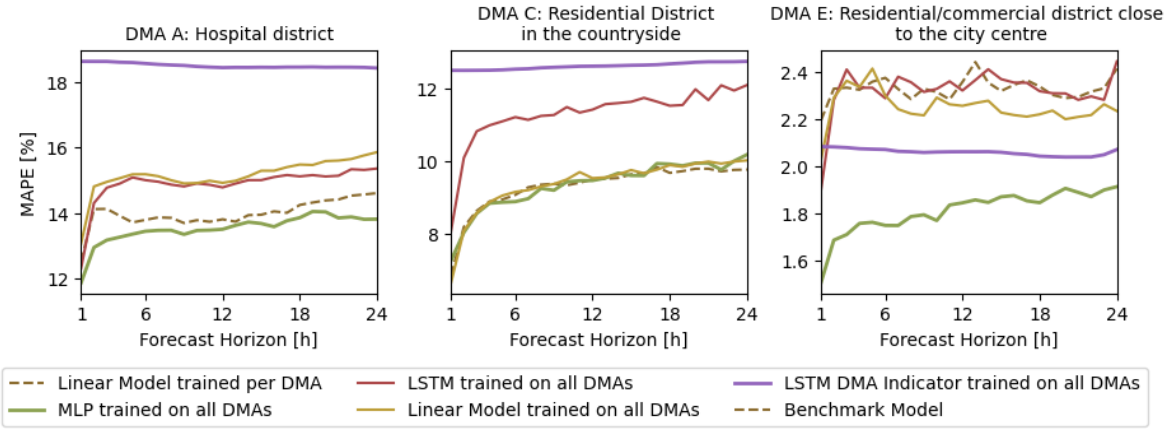
Over the forecasting horizon there are two type of DMAs. DMAs where the benchmark model performs better than the best neural network models at the end of the forecasting horizon model and DMAs where the neural network models outperform the benchmark model entirely. The DMAs with similar results with a better score on the metrics are the residential/commercial district close to the city center (DMA E), the residential district (DMA G) and the city center district (DMA H). These are districts with less annual seasonality compared to all the other districts and with less distribution shifts between both years of data (see Appendix B.4 and B.3). DMAs with more heteroscedasticity over the year have better performing deep learning models.

The error over the forecast horizon is most stable for the benchmark model because this model does take values from a week to four weeks before, thus many of the errors are similar.

In Figure 4.1 DMA A, C and E are shown with MAPE error over the forecasting horizon. It is clear that the LSTM and MLP (trained on a single DMA) models perform signficantly worse while the MLP

model is (similarly as in the plot with all DMAs and models in the Appendix Figure F.3) consistently among the best performing models. The MLP trained jointly over all DMAs performs on average better than any of the architectures over almost the entire forecasting horizon. Compared to the same model with DMA indicator, this difference is small and indiscernible for the first four steps of the forecasting horizon. Compared with any of the other model architectures this difference is much more clear, having a difference of MAPE 0.5 in between. See Figure F.3 in Appendix F.3.

Figure 4.1: Results over forecasting horizon point forecasts of deterministic models on DMA A, C and E (y-axis do not have the same scale)



4.1.2. Results Point Forecasts of Probabilistic Models

Because the deterministic MLP model had the most accurate point forecasts the probabilistic models were based off the MLP model.

The probabilistic forecasting MLP-models with the QR and MDN extension do have a slightly worse performance than the deterministic trained MLP. The Amongst all the MCD has the best performance, on average on all the DMAs, slightly improving the deterministic MLP. The conformalized forecast of the median of QR (model CQR) performs worst among these probabilistic models.

Table 4.6: MAPE [%] Comparison of point predictions of probabilistic models on test set. Deterministic models are indicated with *.

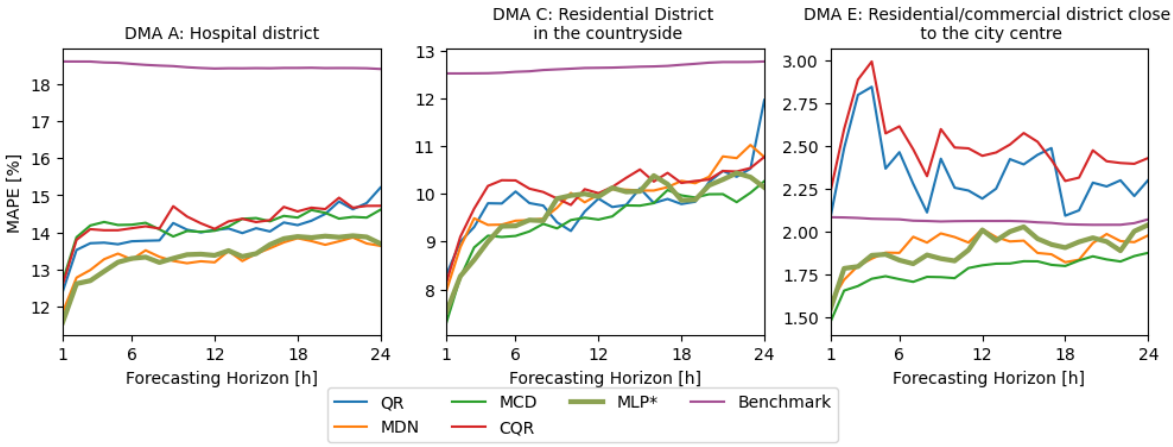
	A	B	C	D	E	F	G	H	I	J	Average
MCD	14.20	**4.65**	9.46	**6.98**	**1.77**	8.97	**3.70**	**4.38**	**5.90**	4.65	**6.46**
MLP*	13.38	4.90	9.70	7.04	1.89	**8.80**	4.01	4.50	6.19	**4.64**	6.51
MDN	**13.37**	4.72	9.90	7.16	1.90	8.86	3.78	4.87	**6.23**	4.71	6.55
Linear model trained per DMA*	13.98	4.87	**9.30**	**7.54**	2.33	8.96	3.91	4.95	6.08	**5.20**	6.71
QR	14.07	5.06	9.85	7.25	2.34	8.90	3.96	5.24	6.15	4.77	6.76
CQR	**14.30**	**5.24**	**10.10**	7.41	**2.50**	**9.09**	**4.22**	**5.35**	6.21	4.95	**6.94**

Table 4.7: RMAE [-] Comparison of point predictions of (mostly) probabilistic models on test set. Deterministic models are indicated with *.

	A	B	C	D	E	F	G	H	I	J	Average
MCD	0.80	**0.70**	**0.72**	**0.87**	**0.87**	0.92	**0.91**	**0.93**	**0.84**	**0.86**	**0.84**
MLP*	0.80	0.73	0.74	0.88	0.91	**0.90**	1.00	0.95	0.89	**0.86**	0.87
MDN	**0.77**	0.71	0.75	0.89	0.92	0.92	0.96	1.06	**0.91**	0.88	0.88
Linear model trained per DMA*	0.80	0.72	**0.72**	**0.94**	1.12	0.93	0.98	1.03	0.87	**0.96**	0.91
QR	0.81	0.75	0.76	0.91	1.13	0.92	0.98	**1.11**	0.89	0.89	0.92
CQR	**0.83**	**0.78**	**0.78**	0.93	**1.20**	**0.94**	**1.04**	1.09	0.89	0.92	**0.94**

Over the forecasting horizon the probabilistic models DMAs E, G and H do have an increase in error in the beginning of the forecasting horizon, around t=3, it is unclear why that is exactly the case. Besides that there are no clear differences. When comparing these average results the MDN, deterministic MLP and MCD have very similar results of which the MCD has a slightly lower error.

Figure 4.2: Results over forecasting horizon point forecasts of deterministic and probabilistic models on DMA A, C and E (y-axis do not have the same scale)



4.1.3. Generalization Results Point Forecasts of Deterministic and Probabilistic Models

Generalization Deterministic Models The linear model trained on all the DMAs as well as the linear model trained per DMA have the lowest GR, meaning the testing error are closest to the validation error. The benchmark model generalizes at the third place (see Table F.7 in Appendix F.1) and after that the MLP trained on all the DMAs. This is important because the MLP model trained per DMA has poor generalization performance (1.68 versus 1.32 on average). Furthermore it shows that the LSTM models have the least best generalization compared to the MLPs and Linear models. On DMA B all models generalize less. Further on DMA E, G and H only the MLP trained per DMA and LSTM trained per DMA have considerably less generalization capability. This highlights that these time-series may have larger differences between the validation and testing series which these models have more difficulty generalizing over.

Table 4.8: Generalization Ratio (GR) of selection of deterministic models

	A	B	C	D	E	F	G	H	I	J	Average
Linear Model trained on all DMAs	1.19	1.62	**1.08**	1.15	**0.99**	1.09	**1.02**	**1.29**	**1.11**	**1.12**	**1.17**
Linear Model trained per DMA	**1.24**	**1.59**	**1.08**	1.13	1.04	**1.08**	1.12	1.35	1.14	**1.12**	1.19
MLP trained on all DMAs	**1.18**	1.87	1.20	**1.12**	1.19	1.10	1.29	1.57	1.28	1.15	1.30
LSTM trained on all DMAs	1.19	2.03	1.29	**1.12**	1.29	**1.16**	1.84	1.91	**1.52**	**1.37**	1.47
MLP trained per DMA	1.23	**2.25**	**1.36**	**1.20**	**1.72**	1.09	**2.76**	**2.43**	1.41	1.21	**1.67**

Generalization Point Forecasts Probabilistic Models In terms of generalization any of the probabilistic models generalize similarly. Small differences are found within the GR between the different probabilistic approaches. As shown in the tables (4.9) the linear models generalize slightly better than the MLP and the probabilistic versions of the MLP generalize slightly better than the deterministic MLP.

Table 4.9: Generalization Ratio (GR) of point forecasts from (mostly) probabilistic models (deterministic models are indicated with *.)

	A	B	C	D	E	F	G	H	I	J	Average
Linear model trained per DMA*	**1.24**	**1.59**	**1.08**	1.13	**1.04**	**1.08**	**1.12**	**1.35**	**1.14**	**1.12**	**1.19**
CQR	1.22	1.82	1.18	**1.15**	1.10	1.10	1.20	**1.35**	1.26	1.14	1.25
QR	1.21	1.85	1.20	**1.15**	1.12	1.10	1.18	1.43	1.30	1.14	1.27
MCD	1.18	1.84	1.22	**1.11**	1.23	1.10	1.29	1.58	1.20	1.13	1.29
MDN	1.18	1.84	1.22	1.13	**1.27**	**1.11**	1.29	**1.79**	**1.34**	**1.17**	1.33
MLP*	**1.17**	**2.04**	**1.25**	1.14	**1.27**	1.09	**1.43**	1.61	1.32	1.14	**1.35**

4.2. Results Prediction Intervals of Probabilistic Models

4.2.1. Calibration Results Prediction Intervals

The coverage for each model and district is computed on the validation set. Ideally, each model is expected to achieve a coverage of 0.95 per DMA, aligning with the target coverage for the forecasts. From Table 4.10 it can be clearly seen, that the target coverage of 0.95 is met for all districts in the CQR and MCD model. The QR model achieves the worst performance with respect to coverage probabilities, only attaining the target coverage in one district (DMA J) and worst target coverages in all the other districts. Comparing the Benchmark model and MDN, the benchmark model performs slightly better. It is important to note that the QR and MDN model have an average coverage of 95 and 96 percent for all districts, closely aligning to the target coverage. This target coverage is not reached when the DMAs are considered on an individual level. Lastly, if we consider the average CG and maximum CG, the CQR and MCD model have the lowest values. The benchmark model and MDN have higher average CG and maximum CG, whereas the QR has the highest average CG and maximum CG.

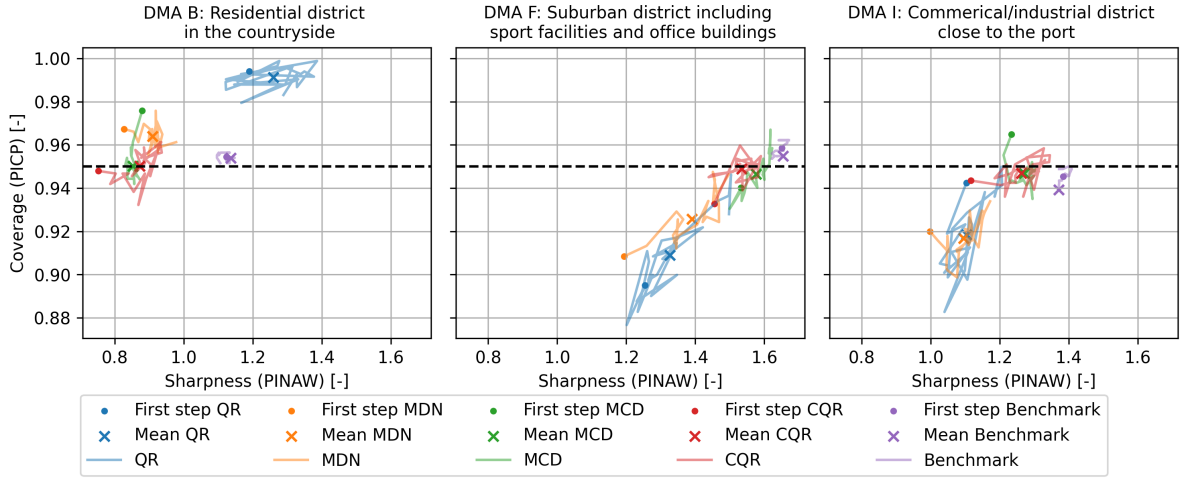
Table 4.10: Coverage probabilities and average CG of models on validation set

	A	B	C	D	E	F	G	H	I	J	Average CG	Max CG
CQR	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.00**	**0.00**
MCD	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.00**	**0.00**
Benchmark Model	**0.94**	**0.95**	0.94	**0.95**	0.97	0.96	0.96	**0.95**	**0.95**	**0.95**	0.01	0.02
MDN	**0.95**	0.96	**0.95**	0.92	0.99	0.93	0.96	0.97	**0.92**	**0.95**	0.02	0.04
QR	**0.94**	**0.99**	**0.97**	**0.91**	**1.00**	**0.91**	**0.98**	**0.99**	**0.92**	**0.95**	**0.03**	**0.05**

In Figure 4.3 we see the coverage vary over the horizon of the probabilistic models. From the previous Table 4.10 we can deduct, that the MCD has on average the desired target coverage of 0.95 over the forecast horizon. However, the figure shows that over the forecasting horizon the average coverage decreases. The MDN and CQR models experience randomness, showcased by fluctuating coverages and not being able to reach the desired coverage of 0.95. The CQR[1] model reaches a coverage close to 0.95. The figure F.5 with all DMAs in Appendix F.5 shows a similar trend for all the other individual DMAs.

[1]The CQR algorithm does not update the residuals on the validation dataset because these are already present within this set.

Figure 4.3: Coverage versus sharpness of probabilistic models on validation set on DMA B, F and I



4.2.2. Results Prediction Intervals

In Table 4.11 we see the coverage of the different probabilistic models[2] on each DMA. On Average the CQR model is best at keeping the coverage to the desired probability of 0.95. Besides the benchmark model the two learned models MCD and QR have a worse performance where the coverage varies from 0.89 to 0.99 for both models. Interesting is that when taking the average of these coverages, these two models have an average coverage of at least 0.93 and 0.94 respectively. In Appendix A the PICP is shown of the CQR model that does not update the residuals where the quantiles are re-calibrated with. It shows for any DMA besides DMA J that the online updating substantially improves the coverage over the test set. In DMA J it is similar. On average the online updating improves the coverage with 3 percent difference.

Table 4.11: Coverage of models across DMAs A to J and average CG on test set, PICP [-]

	A	B	C	D	E	F	G	H	I	J	Average CG	Max CG
CQR	**0.95**	**0.94**	**0.94**	0.94	**0.95**	**0.95**	0.98	**0.95**	**0.95**	**0.94**	**0.01**	**0.03**
MCD	**0.95**	0.85	0.92	**0.95**	**0.90**	0.96	**0.96**	0.89	**0.95**	**0.94**	0.03	0.10
QR	0.92	**0.94**	0.93	**0.89**	0.99	**0.90**	**0.99**	0.96	0.90	**0.94**	0.03	0.06
MDN	0.94	0.92	0.91	**0.89**	0.98	0.91	**0.99**	0.93	**0.87**	0.92	0.04	0.08
Benchmark Model	**0.91**	**0.83**	**0.90**	0.94	0.94	0.93	**0.96**	**0.88**	**0.87**	**0.87**	**0.05**	**0.12**

Analyzing the sharpness of the models in Table 4.12, the MDN is on average the sharpest performing model. MCD is sharpest on most DMAs. CQR is clearly the least sharp model with an average difference of 0.12 in the PINAW score with the MDN model and slightly less sharp than the QR model.

Table 4.12: Sharpness of models across DMAs A to J on test set, PINAW [-]

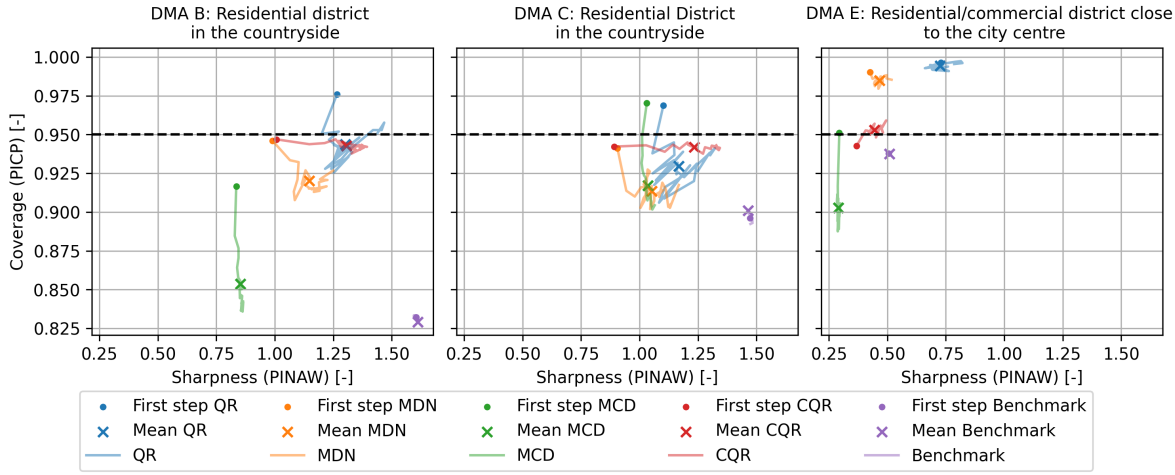
	A	B	C	D	E	F	G	H	I	J	Average
QR	1.35	1.30	1.17	**0.92**	**0.72**	**1.44**	**1.32**	**0.61**	1.21	0.88	1.09
MDN	**1.32**	1.15	1.05	**0.92**	0.47	1.47	1.18	0.46	**1.12**	**0.84**	**1.00**
MCD	1.42	**0.85**	**1.04**	1.07	**0.29**	**1.89**	**0.79**	**0.36**	**1.58**	0.88	1.02
CQR	1.49	1.31	1.24	1.02	0.44	1.73	1.03	0.50	1.51	0.91	1.12
Benchmark Model	**1.51**	**1.61**	**1.48**	**1.10**	0.50	1.80	1.01	0.42	1.35	**0.93**	**1.17**

Figure 4.4 shows the coverage plotted versus the sharpness over the coverage. The first prediction

[2]Note for clarity: all the neural network models discussed here are trained on all DMAs jointly. The probabilistic extensions are used in combination with the MLP because this model had the best average point predictions.

step is marked with a dot and the means are denoted with a cross. By analyzing the trade-off between the coverage and sharpness over the forecast horizon, it shows that the the QR and MDN models have more or less random pattern. This not holds only for the three DMAs shown in Figure 4.4 in Appendix F.5 but for all. For DMA D, H, I and J the randomness can more clearly be observed for the coverage, whereas for DMA B and G this is more clearly visible for the sharpness. On DMA E the models experience minor of change for these two metrics. The MCD model reduces over the forecasting horizon mostly in terms of coverage and has little change of sharpness. The CQR effectively re-calibrates the prediction intervals (of QR) to have a matching coverage but this results in less sharp intervals. The benchmark model performs more poorly on the testing set, with PICP values dropping as low as 83 percent.

Figure 4.4: Results over time horizon probabilistic models on test set with PICP and PINAW on DMA B, C and E



If we accept a degradation of the coverage from 0.95 to 0.93 and select the model with the best Winkler score, the following results are attained.

Table 4.13: Model Performance Across DMAs. Conditional Relative Winkler Score where $\text{PICP}_{\text{testing set}} > 0.95 - \eta$, where $\eta = 0.02$

	A	B	C	D	E	F	G	H	I	J
QR	-	**0.61**	-	-	**1.18**	-	**1.21**	**1.05**	-	0.77
MDN	**0.85**	-	-	-	**0.79**	-	1.09	-	-	-
MCD	**1.09**	-	-	0.92	-	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	**0.60**	**0.72**	**0.91**	0.81	**0.92**	0.97	**0.91**	**0.80**	**0.78**
Benchmark Model	-	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

The results show that the CQR model has the best performance on six DMAs. The MCD model has the best performance on two DMAs and MDN on the three DMAs. The QR and benchmark models have the lowest performance and do not perform best at any DMA.

Table 4.14: Model Performance Across DMAs. Conditional Relative Winkler Score where $\text{PICP}_{\text{testing set}} > \text{PICP}_{\text{validation set}} - \eta$, where $\eta = 0.02$

	A	B	C	D	E	F	G	H	I	J
QR	0.93	-	-	0.94	**1.18**	0.90	**1.21**	-	-	0.77
MDN	**0.85**	-	-	-	**0.79**	**0.88**	1.09	-	-	-
MCD	**1.09**	-	-	0.92	-	**0.98**	**0.81**	-	**0.84**	**0.76**
CQR	1.07	**0.60**	**0.72**	**0.91**	0.81	0.92	0.97	**0.91**	**0.80**	**0.78**
Benchmark Model	-	-	-	**1.00**	-	-	1.00	-	-	-

As mentioned before the QR and MDN models were difficult to calibrate on the validation set. Assume an acceptable drop of 0.02 PICP relative to the validation on the testing set to calibrate the models. Now the results are still in favor of the CQR model where it performs best on five DMAs, the MCD best on two DMAs and the MDN is best on three DMAs. Tables with Conditional Winkler Scores for an η that varies between 0.01 to 0.05 are in F.4.3, including for the full table with Winkler Scores.

The best models according to the Winkler Scores with coverage drops of 0.01 to 0.05 in steps of 0.01 are counted in Table 4.15. It shows that the CQR model performs very well when accepting a low drop in coverage to 0.94 or 0.93 where this model performs best at six DMAs. With a larger drops in coverage up to 0.90, the MDN model performs a better with an acceptable drop to a 0.92 coverage probability at five DMAs, where the CQR model is best at four DMAs. At 0.91 and 0.90 the MDN is respectively best at seven and six DMAs. When allowing a drop relative to the coverage of the validation set, the CQR model performs the best when accepting a drop with η of 0.01, 0.02 and 0.03. For η 0.01 and 0.02 the MDN is better at one more model when counting the models relative to the coverage of the validation data. At larger coverage drops of 0.04 and 0.05 the MDN is better at respectively five and seven DMAs.

Table 4.15: Count of DMAs where each model has the lowest CWS, the coverage decline is relative to validation set and to the 0.95 threshold (η is the decline in coverage)

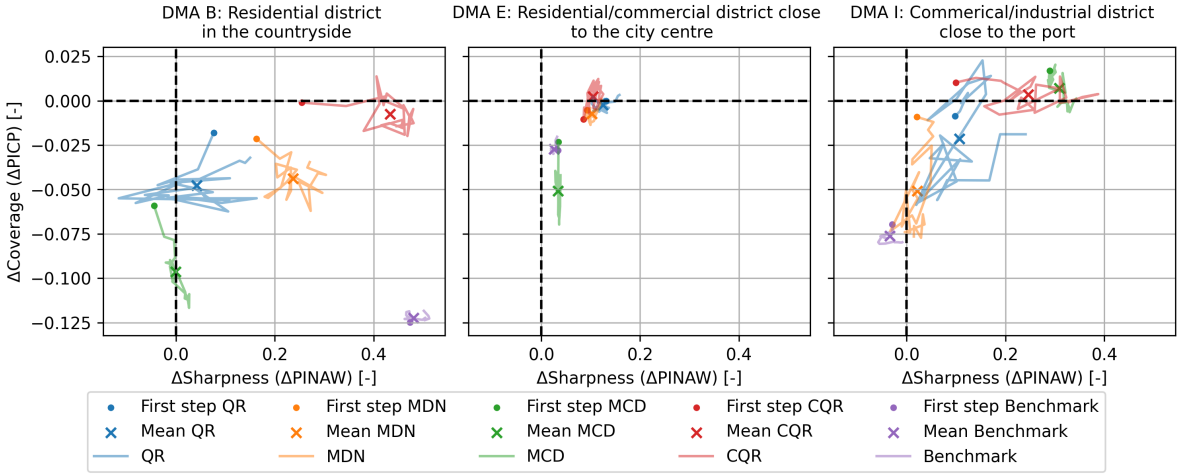
	Coverage Decline Relative to 0.95					**Coverage Decline Relative to Validation Set**				
	$\eta = 0.01$	$\eta = 0.02$	$\eta = 0.03$	$\eta = 0.04$	$\eta = 0.05$	$\eta = 0.01$	$\eta = 0.02$	$\eta = 0.03$	$\eta = 0.04$	$\eta = 0.05$
QR	0	0	0	0	0	1	0	0	0	0
MDN	1	2	**5**	**7**	**6**	2	3	4	**5**	**7**
MCD	3	2	1	1	2	3	2	1	1	1
CQR	**6**	**6**	4	2	2	**4**	**5**	**5**	4	2
Benchmark Model	0	0	0	0	0	0	0	0	0	0

4.2.3. Generalization Results Prediction Intervals of Probabilistic Models

The coverage of the MCD reduces for all DMAs over the forecasting horizon when moving from the validation to test set, but the sharpness does barely change. Note that it is important to understand that the sizes of the validation set for interpretation of this split are much smaller than the testing set and vary. In the discussion 5.1 there is reflected back on this.

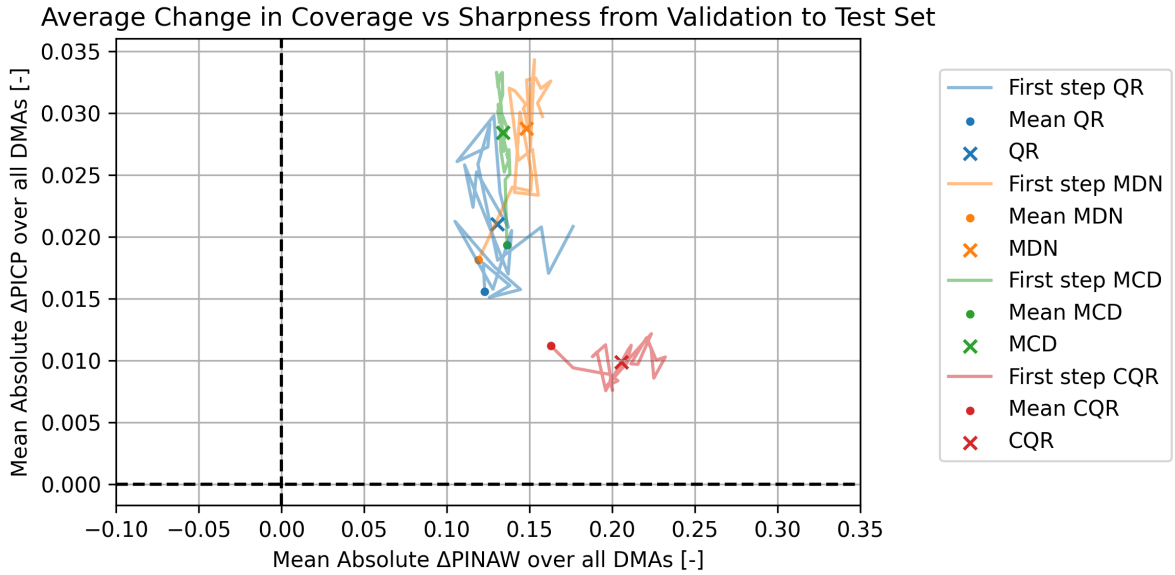
For CQR the PINAW reduces but the coverage is most similar compared to the other models. The QR and MDN models do not have a clear deviation in terms of coverage or sharpness when moving from the validation to test phase/set. This is visible in Figure 4.5 where the deivation of coverage and sharpness are plotted for DMA B, E and I.

Figure 4.5: Generalization of Probabilistic models with ΔPICP and ΔPINAW



When analyzing the average differences of PICP and PINAW between both validation and testing set, it is again evident that the CQR model shows the least change in coverage across all districts when transitioning from validation to testing data, but also has the largest increase of sharpness. The other models have similar absolute differences in terms of sharpness between the validation and testing set, which are less than the CQR model. The absolute differences of the coverage between the validation and test set are besides the CQR the lowest for the QR and then similar for the largest for the MCD and CQR models.

Figure 4.6: Average absolute ΔPICP and ΔPINAW over all DMAs

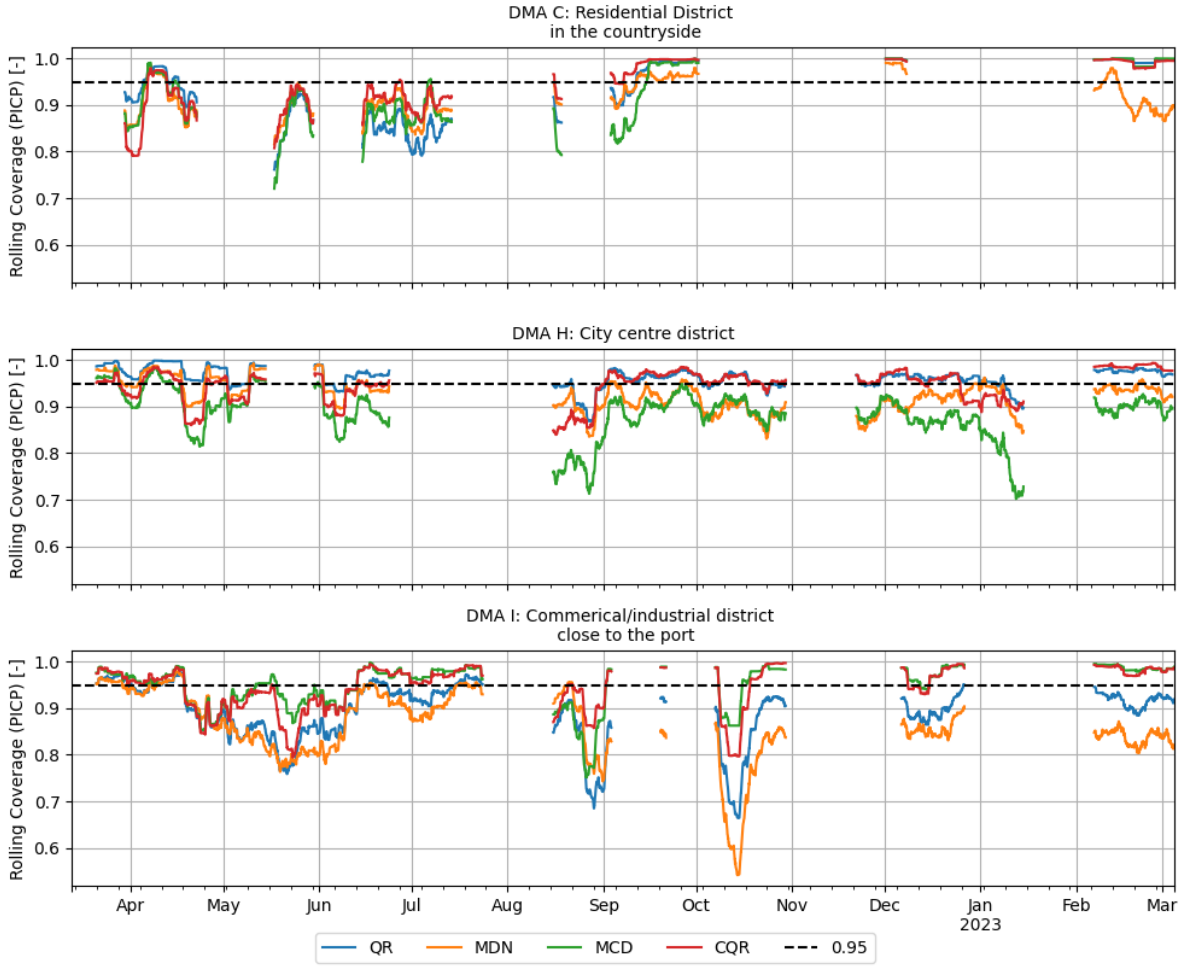


4.2.4. Coverage Over Time

Below in Figure 4.7 the rolling PICP is shown for three DMAs. These DMAs were selected because they had more data without gaps than other DMAs, making it possible to analyze rolling coverages. We see the coverage varies over time, sometimes reaching a low value of less than 0.6 as we can see for the MDN model at DMA I. It seems that the models balance the under-coverage with over-coverage to obtain the desired 0.95 coverage. Some of the declines of coverage do overlap between these

selected DMAs, for example Between DMA C and H when April starts and between DMA H and I in the end of September. Also the other DMAs experience some undercoverage in the spring and summer periods which is shown in Appendix F.5.1. Alternatively, if we compare DMA H and I in mid-October, the coverage declines drastically in DMA I but remains stable in DMA H. In DMA I it is difficult to understand why the model exactly results in lower coverage. When manually analyzing these forecasts any of the models result in predicting too little water demand.

Figure 4.7: Results rolling PICP of each week (168 forecasts) on testing set for DMA C, H and I



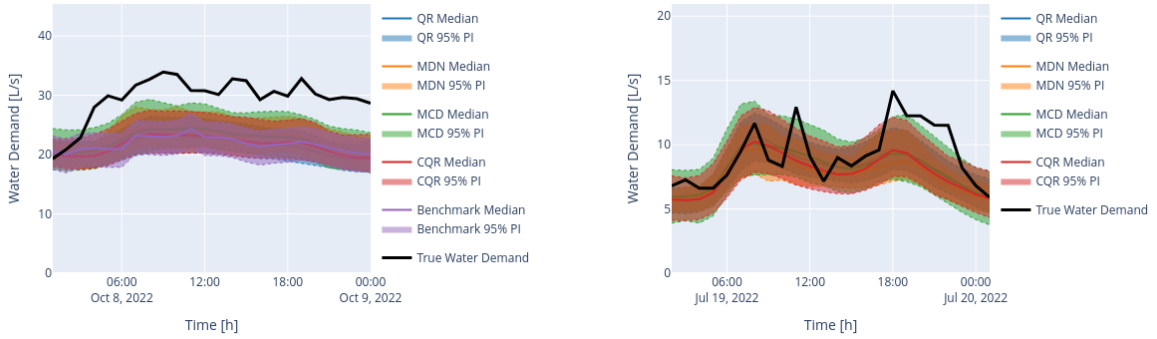
The benchmark model was excluded of this figure because it undercovers more than any of the neural network models making the figure unreadable. Of this model the coverages are for some DMAs sometimes close to 0.4. Especially at DMA B, D and I the benchmark model underperforms. In Appendix F.5.1 besides the figures of the rolling coverages on all DMAs the minimum, maximum, median and average statistics per DMA are given too. Taking the minima from each DMA, the CQR model is on average and median closest to the desired 0.95 coverage. This is a 0.11 average minimum coverage gap per DMA. The lowest the rolling coverage of the CQR ever goes is 0.79. Over all DMAs the QR, MDN and MCD reach coverages as low as respectively 0.66, 0.54 and 0.68. In terms of maximum coverage every model overcoverages with a maximum rolling coverage of 1. The median and mean results show a similar result as the results in Subsection 4.2.2; the CQR model is most reliable.

The probabilistic prediction methods do have some miscalibration between the weekends and weekdays. On average the QR and CQR model do have the lowest difference between the coverage in the weekend and in weekdays. The maximum deviation between weekend and weekday is largest for the MDN model. The lowest maximum deviation is 0.02 for the QR model.

Table 4.16: Difference in PICP Between Weekend and Weekday Across DMAs ($PICP_{Weekend} - PICP_{Weekday}$)

	A	B	C	D	E	F	G	H	I	J	Average Abs. Difference	Max Abs. Difference
QR	**0.01**	**-0.02**	**-0.01**	0.02	**-0.00**	0.01	**-0.00**	0.01	0.02	0.02	**0.01**	**0.02**
CQR	0.02	**-0.01**	**0.01**	0.01	-0.03	0.02	**-0.01**	-0.02	**0.01**	**0.00**	**0.01**	0.03
MCD	0.02	**-0.02**	**0.01**	**0.00**	**-0.05**	0.01	**-0.01**	**0.00**	0.02	0.01	**0.02**	0.05
Benchmark Model	**0.03**	**-0.01**	**0.00**	0.01	-0.02	**0.00**	**0.00**	**0.03**	0.06	**0.04**	**0.02**	0.06
MDN	**0.01**	**-0.02**	**0.01**	**0.03**	-0.01	**0.03**	**0.00**	0.02	**0.08**	0.01	**0.02**	**0.08**

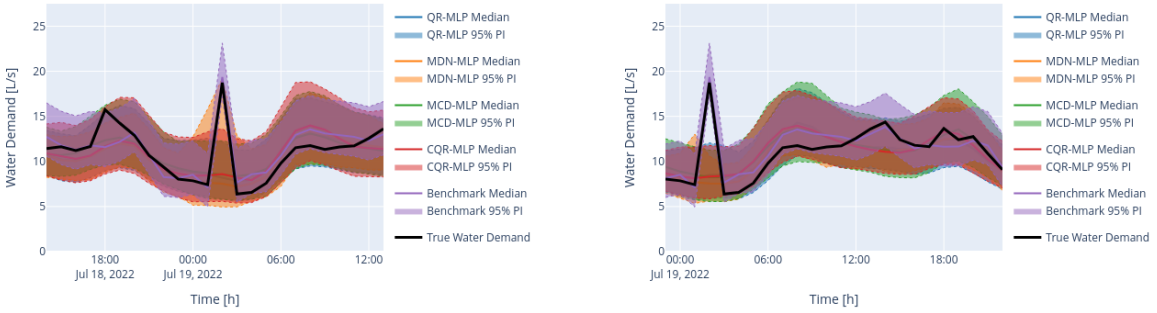
Figure 4.8: Example of undercoverage in DMA A



(a) Mismatch in trajectory

(b) Mismatch in combination of trajectory and peak

Figure 4.9: Example of peak in DMA A



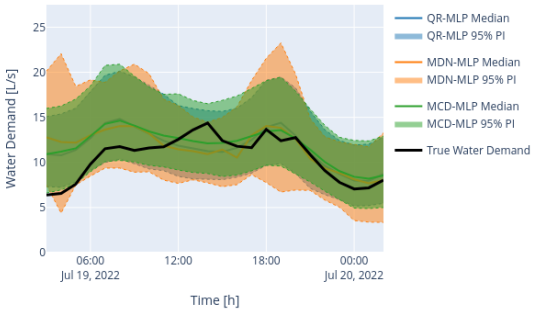
(a) Peak only captured by MDN and Benchmark model in DMA A

(b) Peak not captured by MDN.

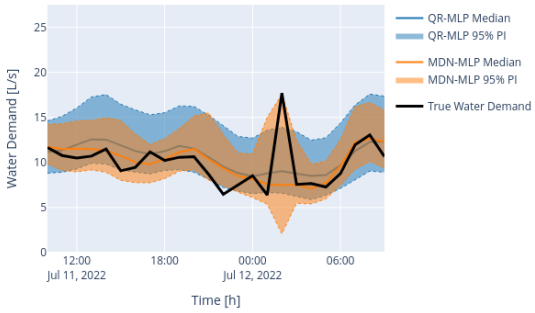
DMA A has peaks in the water demand scheme, with some temporal variation throughout the year. These occur mostly in the beginning of 2021 on Tuesdays at 02:00 AM, then in November 2021 these are as well at Mondays at 00:00 AM. In April 2022 these switch to Sunday 11:00 PM and Tueday 01:00 AM. These are missed for the median model results by any model besides for the benchmark model, the models QR, MCD and CQR do miss these as well in the uncertainty bounds.Only the MDN-MLP model increases the prediction interval to capture these results, but this does not occur consistently and often fails to extend the interval sufficiently. The benchmark model does capture it, unless it changes the time because then the average of the previous weeks does not match the new pattern. It is interesting that the MDN sometimes captures this as uncertainty because this model as well as the quantile regression model are both trained similarly. This shows that the MDN method is in general more sensitive to learn heteroscedasticity. However, it is not perfect. the median ignores these peaks as well as the heteroscedasticity peaks down as well.

An effect of the peaks in DMA A are that it disrupts the models. The prediction intervals become excessively wide, and the median predictions overestimate the first few hours of the forecasting horizon.

Figure 4.10: Example of peak in DMA A



(a) Increase of width right after peak



(b) Downward Peak Prediction Interval MDN

5. Discussion and Recommendations

The discussion is presented in Section 5.1 which is divided in Subsection 5.1.1 covering the data splits, Subsection 5.1.2 discussing the training method and 5.1.3 examining the models. Lastly, the recommendations of this research are provided in Section 5.2.

5.1. Discussion

5.1.1. Discussion Data Splits

Additional Data Splits Additional data splits would have been useful, since the validation data is used to save the best model during training. This validation data split is also utilized to calibrate the prediction intervals of the CQR and MCD models, causing a slight bias in the results because the models would perform slightly better on the validation split than a a potential second data split. Having an extra data split for calibration purposes would have decreased such bias. This is challenging with the current dataset because it only consists of two years and two months. Ideally, each split has the same distribution which requires the time-span of the longest seasonality to be in this split. Adding an extra split consisting of slices, similar to what was done with the validation data, was challenging because it would have removed a significant amount of training data. As discussed in paragraph 5.1.3, this limitation has already posed challenges in DMA B. The decrease of data was not only due to the slices themselves but also because of the padding required to ensure no overlap between the different data sets. Defining the splits of the current validation set already posed a challenge because it required to search for splits that generate sequences for each of the ten districts at the same time, within the first year and two months. Having these splits in the second year would make it a lot more difficult to assess the generalizability, as the distribution between validation and testing splits would be too similar. If the splits were set at different points in time across DMAs, or if one DMA was used exclusively as validation data, this could result in data leakage. This would entail that the trained model would be biased and has improved forecasts due to correlations between the districts in training, validation and testing data. Alternatively, not all ten DMAs need to be evaluated, thus if only few DMAs would be excluded it still would be acceptable.

Cross-Validation Alternatively a cross-validation approach could have been used where the model is trained (including hyperparameter optimisation) and calibrated multiple times, each time on different folds (subsets) of the training data. Then an aggregated ensemble of these models could have been run on the testing period. The downside of this is that the annual seasonalities of DMA A, B and C would surpass the length of each cross validation fold. This could potentially decrease the model performance because it would not have been exposed to the full distribution of water demand. Additionally, it would also increase the training- and execution times significantly.

Interpolation To maximize the number of sequences, the first year and two months were interpolated, limited to a step size of 3. This was the data of the training and validation splits and was done per section of this data. This introduces some bias to the model, however it is expected to be minimal due to the limited step size. With the current method, the number of training and validation sequences generated from the data was quite low prior to interpolation. Ideally, the validation data should not be interpolated.

5.1.2. Discussion Training Method

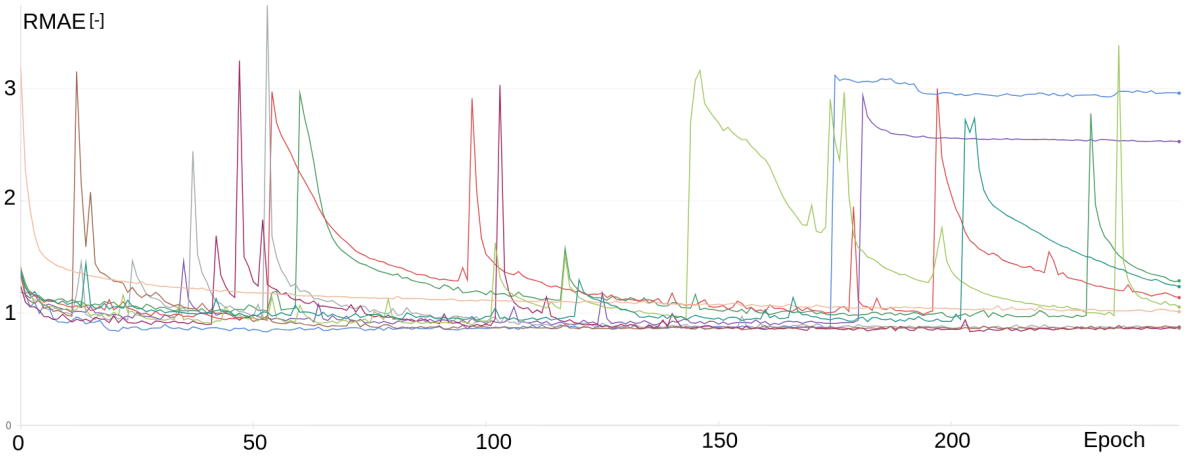
Model Saving Criterion on the Validation Set The deterministic models are saved every epoch with the best average RMAE over all DMAs. This forces the model to have on average a better performance than the benchmark model. The deterministic models do not perform a lot better than the benchmark model on the DMAs that have a more repetitive pattern. Investigating alternative metrics that do not depend on a benchmark model would be interesting, since this could potentially yield different results. Perhaps the model performance on more challenging DMAs (such as A, B, and C) could be improved

by using a metric that specifically encourages the models to perform better on these districts. This can for example be done by using a normalized form or a weighted form of MAE.

The QR and MDN had slightly worse point predictions compared to the deterministic MLP. One explanation could be that when training the QR and MDN models, the best model was saved with the best RQE on the validation data during training, which is the Quantile Error (mathematically same as Quantile Loss) divided by the Quantile Error of the benchmark model. This metric uses the median (point prediction) as well as the interval predictions. This causes the model to learn a model that was relatively better than the benchmark model across all DMAs (on the validation data), relative to the point and interval forecasts according to this metric. It could be that the point predictions from the probabilistic models are slightly worse than the ones of the deterministic model because this metric negatively influences the point forecasts. To accurately determine this, the model should be trained by selecting the best model based on a criterion applied to the validation data, which divides the Quantile Score of the probabilistic model by the RMAE of the benchmark model.

Training the LSTM When training the LSTM model, the exploding gradient problem was encountered which means that the errors abruptly increase. This could be one of the reasons the LSTM did not perform adequately. For further studies involving LSTM, the sLSTM, mLSTM, or xLSTM variants can be considered. These models incorporate improved gating mechanisms designed to enhance stability during training [7].

Figure 5.1: Exploding Gradient Problem of LSTM. Each line is the RMAE per epoch, evaluated on the validation set.



5.1.3. Discussion Models

Linear Model The linear model is mathematically similar to a linear regression, expressed as $\hat{y} = w_0 + w_1 x + w_2 x + \cdots + w_{168} x$. In this research both linear models performed similarly. One was trained per DMA and one jointly. The minor differences in results can firstly be explained by variations in the distributions between the two datasets (Appendix B.4) and differing autocorrelation between the different DMAs (Appendix B.4.1). Furthermore, the linear model has a hidden size of 128 when trained per DMA. In initial experiments, the model was trained with a hidden size of 24 to maintain mathematical similarity with the expression above, however this configuration resulted in a poorer performance. The exact reason for this outcome is unclear. Somehow the training process was able to find better fitting parameters with a larger hidden size even though there is no mathematical advantage. In retrospect even larger hidden sizes could have been explored by the Bayesian optimization.

Epistemic Uncertainty of Probabilistic Models It was assumed that the model, whether it did or did not utilize the DMA indicator, would also perform best for the probabilistic prediction. In hindsight, this seems unlikely for the QR and MDN model because the results clearly the presence of deviations in

terms of coverage between the weekends and weekdays and between DMAs. In other words, there is still epistemic uncertainty left to reduce, for example with categorical values or if we assume a perfect weather forecast. If more features would have been added, the MLP could have issues with curse of dimensionality. This refers to the exponential increase in data required to reliably train models as the number of features grows, leading to sparsity and making it harder to capture meaningful patterns. Thus it is not likely the best model architecture in its current form. Also modeling heteroscedasticity including the point forecasts (median) is a more complex problem, which could mean some architectures could perform better for point forecasts and others better for probabilistic predictions.

Number of Samples MCD, MDN and CQR When fitting the MCD and MDN models for both 1000 samples were used for respectively the forward passes and the number of times that are drawn from the parameterized Gaussian mixture distribution. This could have been attributed to overparametrization, however such overparametrization does not negatively affect the results. During the process a lower number of draws were initially used of 100 and 300. When re-doing this with different seeds the coverages on the test set did change. With 1000 samples this was not/barely the case. A potential overparametrization does not negatively influece the results. For any conformal prediction models there are ideally 1000 samples according to [2], but the validation data used to re-calibrate the quantiles have around 700 to 800 sequences. Nevertheless, the CQR was still the most reliable model in terms of performance, which can be attributed to online-updating. This process results in an increase of the sample size and the distribution is updated accordingly.

Effect of Online Updating CQR In Appendix A a small experiment is shown where the coverage of the CQR model is compared to a scenario, where online updating of the recalibration process was disabled. This resulted in both the CQR and MCD having a coverage of only 0.85. The training, validation and testing data in Appendix E shows that from May to July 2021 there is hardly any data in DMA B for the validation split. In the testing data this is timeframe that has the highest concentration of data in 2022. This likely explains why the MCD model (and the CQR that does not use online updating) have low coverage on the testing set. This problem is also encountered in other DMAs, however identifying this is more difficult due to the similarity of the data and results. Lastly it is important to mention, that the different sizes of the validation splits did not have a noticeable effect on the results of the models.

Benchmark Model The benchmark model, which was fitted per month, has noticeably lower coverages for DMA B, I and J. The low coverages are likely due to the differences between the monthly distributions of different months within the first and second year of the data. The benchmark model uses each month of the first year to construct the prediction intervals. Furthermore, this also affects the Normalized Winkler Scores, which were calculated by dividing the Winkler Scores of the Neural Networks by the Winkler Scores of the Benchmark model.

5.2. Recommendations

Repetition of Experiments The current experiments have been conducted once. Even though the Bayesian optimization is conducted with 100 model runs, it is important to be aware of the standard deviation of the results. Thus it may be interesting to re-do the hyperparameter optimization repeatedly and evaluate how the results differ between the best models of these hyperparameter optimizations.

Instead of Interpolation A Masked Loss The maximum interpolation step of the training and validation data was defined to be 3 due to the large number of gaps in the time series and their short length. Data sequences were generated by skipping sequences with gaps, and the models were trained on these sequences (168 input values and 24 output values), assuming the interpolation bias is negligible on the validation and training set. For future research, it may be valuable to retain n gaps in the ground-truth sequences while masking the gaps during training. This approach would prevent bias when updating model parameters. Additionally, this setup would improve the validation process, as more sequences could be used without interpolation. Incorporating the number of interpolation steps (and n) as a hyperparameter, could also improve the training of the models.

More Variables For point forecasts the DMA-indicator did not have any influence in improving the point forecasts. Perhaps the combination of more temporal variables (month of the year, day of the week, hour of the day et cetera), national holiday data and weather variables could have a positive impact on the point forecasts. This can be explained by the different effects that weather and holidays have on different DMAs at different moments in time. We know weather effects human water use [58] thus it is interesting to improve the forecasts of non-industrial DMAs with. Categorical variables can be added using a one-hot encoding, but if we add a variable such as month of the year, it will not work with the current dataset. The validation splits take out almost an entire month of data, keeping some parameters (barely) untrained. Also with only one full year of training data there is only one month of data per category. In such case adding a sine and/or cosine transformation of categorical data may be more beneficial. It is recommended to investigate the impact of these additional variables, for example by adding them to the Bayesian hyperparameter optimization.

DMA A (hospital district) has peaks that change throughout the dataset to a slightly different pattern. This makes them difficult to predict, since it requires knowing at what time they occur in the future. It is possible to assume these peaks occur at the same hours and days as the previous week(s). This can be detected and incorporated as a categorical value using one-hot encoding and is recommended for future forecasting with this dataset.

Recommendation Dataset A different dataset was found at a later stage of this research. This dataset has six years of data from five residential areas in the Netherlands without missing values on a 15-minute timescale. Further research is recommended to use this dataset as it alleviates many of the problems connected with the utilized dataset (link found in reference [39]). Furthermore, it is recommended to make a global dataset of water demand data. Studies can utilize this dataset which can help identifying the results of different forecasting approaches on the same set.

Univariate Models for Point Predictions Because the linear model performed well it is recommended to apply an AR [28] model on this dataset. This is a more common approach for forecasting time-series in an univariate manner than the currently used linear model. A different univariate modelling approach is to expand the benchmark model with weighted moving averages, such as the exponential smoothing model [28] because the benchmark model performed better on DMAs with lower heteroskedasticity. For further studies it can be interesting to use these methods as a benchmark for point predictions.

Multivariate Models When using more input variables it would be interesting to evaluate a LSTM in an encoder-decoder fashion because with this approach it is possible to use future features that have a non-linear relationship with the water demand, as a multivariate model. This makes the LSTM interesting again because when using a MLP for this, the increase of features will make it more difficult to learn this system adequately due to the curse of dimensionality. Any model that sequentially processes data will be favorable from this point of view. With LSTM encoder-decoder models (for example [16] or [47]) there is no information exchange between future input variable relative to the prediction step (x_{t+i} and it does not have an influence on the prediction $\hat{y}_{t+i-1}$). That may not represent reality because human behavior and practices may change with different forecast information. For this an adaptation may be investigated with specific weather embeddings or the use of BI-LSTM encoders. Because these models are more complex and due to the already poor performance of the current LSTM, it may be required to have a lot more data available for training.

If an encoder-decoder model does not provide adequate results, other machine or time series analysis models can be used. These can be random forests and gradient boosting models or models from classical time-series analysis, however they require differencing the data. Differencing is required for tree based models such as random forests or gradient boosting for extrapolating the time series and as well for some classical time series analysis models to remove seasonality [28]. A downside of differencing the data is it propagates the error over the forecasting horizon. For short-term time series forecasting such as in this research may not be a problem but for longer forecasting horizons neural networks may therefore be more suitable.

Calibration of Models The QR and MDN models were difficult to calibrate on the different DMAs because it was trained on all DMAs. It is recommended to evaluate how these two probabilistic extensions perform when trained on top of the linear model trained per DMA. Then it can be analyzed if the coverages are closer to 0.95 per DMA on the validation and testing sets. Alternatively, the losses can be computed per DMA separately (on the model that is trained jointly on all DMAs). Then regularization can be applied separately per DMA. It can potentially help to force this model in having similar coverages for the different DMAs on the validation set. Within the course of this thesis some experimentation was done by saving and storing the model with the lowest coverage gap during training. This was not further analyzed in this research, due to the time intensive nature. This criterion did not improve any interval forecasts and decreased the point forecasts. Using this metric for the Bayesian optimization too would make the prediction intervals wider than the minimum and maxima of the demand pattern.

The MCD model hardly adjusted the width of the interval over the forecasting horizon for some DMA. It is not advisable in the setup of a model, that it is trained solely on lagged water demand on multiple DMAs. Perhaps more features could make a difference for this or having a model trained per DMA. When adding features it is expected that these would need to improve the point forecasts too and not solely explain the heteroscedasticity, because the network somehow would need to learn how to utilize them. MCD does not directly have a way to incorporate features for heteroscedasticity. Furthermore, it would be interesting how different model architecture affect the coverage of the prediction interval over the forecasting horizon because the dropout mechanism is applied on the parameters of the model. More experimentation can also involve using dropouts on specific layers.

Online Recalibration/Training The CQR model did benefit in terms of coverage on the test-set with online updating. The missing data still affects the current performance. Having no missing data could have potentially improved the results. If wanting to use MCD on a model it is advised to analyze to research how online re-calibration can improve the model. The current dichotomic search for fitting the dropout rate did not require more than five tries to find the required dropout rate, which should for many models be efficient enough for online use. This is expected to increase the performance of MCD models. Otherwise these models can be retrained too every couple months. It can be interesting how re-training or continuing the training method can improve these forecasts, especially by comparing this to an online updating method. Further it is recommended to analyze different methods for recalibration, for example by using the past N data or that but with data of years ago added to make sure the distributions of data are similar between the future data and the set to calibrate the models on.

As explained in the discussion for the probabilistic models, the current criterion that is used to save the best model on the validation split does perhaps decrease the model performance of the median predictions. Perhaps this works the other way around too that the criterion degrades the interval forecasts too. One method to overcome this potential problem is to use two models, one to learn the median and one to learn the quantiles relative to the median. The conformal prediction framework allows for this too as a different way to model heteroscedasticity separately. Thus it is recommended to research conformal prediction further because the model (CQR) yielded the most reliable results of the probabilistic forecasts. The residuals of deterministic models can be used to construct a prediction interval too, for example as applied in [60]. The studies [61, 34] use a second model to learn quantiles directly from residuals, effectively learning the levels of heteroscedasticity, but for example [14] divides residuals of the point forecast with a model that is trained to predict the magnitude of these residuals of which the quantiles are constructed. Alternatively, with enough data the residuals can be grouped similarly as done for the benchmark model. Perhaps more historic data of the same months would have been useful, as they are expected to have a similar distribution. Instead of using predetermined categories these groups can be learned with a clustering algorithm such as k-means clustering [10] or with a Modern Hopfield network where instead of a selection an association score is learned which is then used to construct the quantiles [3]. The latter method addresses distribution shifts too. Alternatively, weighted methods can also be used to address the distribution shift. For example the research of [51] uses an upper bound derivation of the distribution-shift between input variables of the training data and recent input of the test variables to adjust the quantiles. The study [62] deals with distribution shift by using a form of online gradient descent to create wider quantiles with more distribution shift. These methods are still researched heavily and according to the authors knowledge there is no conformal prediction method yet that deals with all of the above mentioned aspects for multi-step models. Even combina-

tions of the aforementioned methods are still not investigated and conformal prediction methods are still primarily investigated in single-step methods.

Advantage of Forecasting the Full Distribution In this research the 0.95 prediction interval was constructed by predicting the 0.025 and 0.975 quantiles. When modeling the entire distribution it is possible to construct the prediction intervals via different methods. For example the 0.02 and 0.97 intervals can be selected, or the most risk-averse selection can be made constructing a worse-case prediction interval. Furthermore, this also allows for an extra calibration step that can be performed online as well by selecting a slightly wider interval over time. This can be achieved using the MCD method, MDN method and with conformal prediction methods too, given the quantiles are constructed from the distribution of (non-absolute) errors of point predictions.

Ensembles to Reduce Variance For future research it is interesting to analyze ensembles of probabilistic models to improve prediction quality because they often make the forecasts more accurate by reducing variance [46].

Another Probabilistic Modelling Approach The number of parameters of the MDN increase significantly with the number of Gaussians that are used. This limits the complexity of the parameterized distribution because it will exceed the number of parameters of other neural network approaches to compare with. A different approach is recommended for analysis, which is an uncountable number of Gaussians/Laplacians, that reduce the number of parameters [11, 33].

Benchmark Models of Previous Studies Within the literature many models were suggested and used. It is interesting and recommended for future research to benchmark these models across various water demand datasets. By doing this it will be more evident which type of models will perform better for different types of water demand time series. For this research this was unfortunately out of scope.

Models Without Lagged Values Now almost all models do require inputs of previous demand series to predict the next value(s). In case a previous value is not present due to an error or data quality issues, it is impossible to predict the future water demand. Thus it is interesting to develop a model solely based on exogenous variables.

Adaptation of Coverage Gap Metric When computing the coverage over an entire data split using the PICP, the PICP can balance out under-coverage with over-coverage as found in the results of Subsection 4.2.4. A way to overcome this is to compute the average distance of a rolling PICP to the aimed coverage. The over-coverage distance can be neglected depending by the objective of the forecasts. This metric is effective at finding the most reliable forecasts of large number of forecasts.

6. Conclusion

This research focuses on exploring the potential of different probabilistic forecasting methods for water demand forecasting with a forecasting horizon of up to 24 hours ahead, using different neural network approaches. The research highlights the differences in forecasting accuracy between probabilistic and deterministic models with respect to point forecasts and the performance of probabilistic models with respect to the probabilistic forecasts. The conclusions for each research question are summarized below.

Research Questions About Point Predictions

RQ1: Which deterministic neural network models are most suitable for short-term water demand point predictions?

Comparisons were made between a linear model, a MLP, and a LSTM model against a benchmark seasonal moving average model using data from ten district metered areas (DMAs). Overall, the MLP model trained on all DMAs had on average the best performance (MAPE 6.59, RMAE 0.88) and is thus the most suitable model of all tested models. The linear models trained per DMA (MAPE 6.71, RMAE 0.91) and with all DMAs together (MAPE 6,74, RMAE 0.90) performed slightly worse than the MLP but generalized slightly better (GS linear model on all DMAs 1.17, linear models per DMA 1.19 and MLP 1.30), making them interesting alternatives. For DMAs with little heteroscedasticity the benchmark model performed better over the entire forecasting horizon, except for the first six steps where neural network models performed better, showing that a neural network model does not always perform better.

RQ2: How does the accuracy of short-term water demand forecasting differ between individual models for each district metered area and a single model for all areas?

Both MLP and LSTM models did show on average for all DMAs significant improvements when increasing the training data from a single DMA per model to all DMAs per model. The linear model showed no improvement. The MLP improved on average over all DMAs from a MAPE of 7.73 and RMAE of 1.11 to a MAPE of 6.43 and RMAE of 0.85. It is important to mention that the improvement was found for each DMA. The LSTM improved on average from a MAPE of 8.83 and RMAE 1.26 to a MAPE of 7.50 and RMAE of 1.03. The LSTM trained on each DMA performed better for nine out of ten DMAs. The linear models did perform similarly when comparing between a model trained per DMA or on all DMAs with very low performance differences that are average over all DMAs 0.01 on the RMAE and 0.03 on the MAPE metric.

Research Questions About Probabilistic predictions

RQ3: Which probabilistic prediction methods are most suitable for interval predictions of water demand?

The CQR model had the most reliable performance because the coverage was closest to the target coverage threshold of 0.95. According to the Winkler Score, where the coverage is allowed to drop from the desired 0.95 in steps of 0.01 to 0.90, the CQR performs best for drops of 0.01 to 0.02 relative to the desired 0.95 coverage probability on six out of ten DMAs. Larger drops to 0.90 favor the MDN model. The QR and MCD models were difficult to calibrate on each DMA, thus the Winkler Scores were computed as well for the DMAs where the coverage would drop 0.01 to 0.05 relative to the coverage on the validation set. Here the CQR performed best on five out of ten DMAs when allowing the coverage to drop up to 0.03 relative to the coverage of the same model on the validation set. Coverage drops of 0.04 and 0.05 resulted in the CQR and MDN models to be better at four out of 10 models.

For each DMA there are different models that performed best. In most cases this was the CQR. If reliability is preferred this model was deemed the most accurate. If a sharper prediction interval is preferred the MDN-model performs second best. Furthermore, the MDN was able to adapt the prediction interval for peaks that were not explicitly used as inputs in the model, but not all the time. The other models missed this.

RQ4: How do probabilistic models compare to deterministic models in terms of point prediction accu-

racy for prediction of water demand?
The point prediction accuracy of the deterministic MLP model is MAPE 6.51 and RMAE 0.84. The QR, MDN and CQR performed less accurate with respectively MAPE scores averaged over all districts of 6.74, 6.76 and 6.94. The respective RMAE errors were 0.92, 0.88 and 0.94. The MCD model performed the best with a MAPE of 6.46 and a RMAE of 0.84, which is slightly better than the deterministic MLP model. The MDN performed only better at DMA A, as compared to every other model based on both metrics.

Research Questions About Generalization
RQ5: What is the performance difference between the validation phase and the test phase of the models?
Between the validation and test phases for point forecasts the deterministic linear models generally showed the best generalization when comparing the GS. The LSTM models generalized the poorest, with the LSTM trained per DMA performing the worst, followed by the one with the DMA indicator, and then the one without it. The MLP trained per DMA did not generalize well, particularly for three specific DMAs. The deterministic MLP trained on all DMAs had the second-best generalization among deterministic models. Among probabilistic models for point predictions, the probabilistic variants of the MLP generalize slightly better than the deterministic trained MLP. The probabilistic models have varying degrees of generalization capability in terms of coverage and reliability. The MCD decreases in coverage while the CQR model increases the sharpness. The MDN and QR models do not have a clear pattern in terms of generalization per DMA, but on average the QR model generalizes slightly better than the MDN model.

RQ6: When do the probabilistic models exhibit under/over-coverage?
The probabilsitic models undercover more during the spring and summer periods. Sometimes under-coverage is more DMA specific, for example in October in DMA I. The models seem to balance these under-coverages with over-coverages during other moments in time. Note it is difficult to compare with every district due to missing data. There are still miscalibrations when analyzing the models in the weekend and the weekday, resulting in different coverages between them. The QR and CQR model have the lowest average difference in PICP between weekdays and weekends.

References

[1] S. Alvisi and M. Franchini. “Assessment of predictive uncertainty within the framework of water demand forecasting using the Model Conditional Processor (MCP)”. In: *Urban Water Journal* 14.1 (Jan. 2017), pp. 1–10. ISSN: 17449006. DOI: 10.1080/1573062X.2015.1057182. URL: https://www.tandfonline.com/doi/abs/10.1080/1573062X.2015.1057182.

[2] Anastasios N. Angelopoulos and Stephen Bates. “Conformal Prediction: A Gentle Introduction”. In: *Foundations and Trends® in Machine Learning* 16.4 (2023). ISSN: 1935-8237. DOI: 10.1561/2200000101.

[3] Andreas Auer et al. “Conformal Prediction for Time Series with Modern Hopfield Networks”. In: (Mar. 2023). URL: http://arxiv.org/abs/2303.12783.

[4] Martijn Bakker. “Optimised control and pipe burst detection by water demand forecasting”. In: ().

[5] Mo’tamad Bata, Rupp Carriveau, and David S.-K. Ting. “Short-term water demand forecasting using hybrid supervised and unsupervised machine learning model”. In: *Smart Water* 5.1 (Dec. 2020). DOI: 10.1186/s40713-020-00020-y.

[6] *Battle of Water Networks | 3rd WDSA/CCWI Joint Conference*. URL: https://wdsa-ccwi2024.it/battle-of-water-networks/.

[7] Maximilian Beck et al. “xLSTM: Extended Long Short-Term Memory Memory Cells xLSTM Blocks xLSTM mLSTM + Exponential Gating + Parallel Training + Covariance Update Rule + Matrix Memory LSTM + Exponential Gating”. In: (2024).

[8] Springer-Verlag Berlin et al. *NATO ASI Series Advanced Science Institutes Series*. Tech. rep.

[9] Christopher M Bishop. “Mixture Density Networks”. In: (1994). URL: http://www.ncrg.aston.ac.uk/.

[10] Henrik Boström et al. “Mondrian Conformal Regressors”. In: *Proceedings of Machine Learning Research* 128 (2020), pp. 1–20.

[11] Axel Brando. *Aleatoric Uncertainty Modelling in Regression Problems using Deep Learning*. Tech. rep.

[12] Bruno M. Brentan et al. “Hybrid regression model for near real-time urban water demand forecasting”. In: *Journal of Computational and Applied Mathematics* 309 (Jan. 2017), pp. 532–541. ISSN: 0377-0427. DOI: 10.1016/J.CAM.2016.02.009.

[13] Joseph Cahill et al. “COVID-19 and water demand: A review of literature and research evidence”. In: *Wiley Interdisciplinary Reviews: Water* 9.1 (Jan. 2022), e1570. ISSN: 2049-1948. DOI: 10.1002/WAT2.1570. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/wat2.1570%20https://onlinelibrary.wiley.com/doi/abs/10.1002/wat2.1570%20https://wires.onlinelibrary.wiley.com/doi/10.1002/wat2.1570.

[14] Nicolo Colombo et al. *On training locally adaptive CP*. Tech. rep. 2023, pp. 1–15.

[15] Diego Corredor. *Short-term Water Demand Forecasting at a District Level Using Deep Learning Techniques*. 2021. URL: https://repository.tudelft.nl/islandora/object/uuid%3A0e79007c-5bb7-4ac7-9dfe-d15dade6e202.

[16] Jing Deng et al. “Operational low-flow forecasting using LSTMs”. In: *Frontiers in Water* 5 (2023). ISSN: 26249375. DOI: 10.3389/frwa.2023.1332678.

[17] Emmanuel A. Donkor et al. “Urban Water Demand Forecasting: Review of Methods and Models”. In: *Journal of Water Resources Planning and Management* 140.2 (Feb. 2014), pp. 146–159. ISSN: 0733-9496. DOI: 10.1061/(ASCE)WR.1943-5452.0000314/ASSET/8DAD8EAD-6B83-476F-B684-0B04F6EA374E/ASSETS/IMAGES/LARGE/FIGURE2.JPG. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29WR.1943-5452.0000314%20https://ascelibrary.org/doi/10.1061/%28ASCE%29WR.1943-5452.0000314.

[18] Ömer Esen, Durmuş Çağrı Yıldırım, and Seda Yıldırım. “Threshold effects of economic growth on water stress in the Eurozone”. In: *Environmental science and pollution research international* 27.25 (Sept. 2020), pp. 31427–31438. ISSN: 1614-7499. DOI: 10.1007/S11356-020-09383-Y. URL: https://pubmed-ncbi-nlm-nih-gov.tudelft.idm.oclc.org/32488700/.

[19] Francesca Gagliardi et al. “A probabilistic short-termwater demand forecasting model based on the Markov chain”. In: *Water (Switzerland)* 9.7 (July 2017). ISSN: 20734441. DOI: 10.3390/w9070507.

[20] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning”. In: *33rd International Conference on Machine Learning, ICML 2016*. Vol. 3. 2016.

[21] Tilmann Gneiting and Adrian E. Raftery. “Strictly proper scoring rules, prediction, and estimation”. In: *Journal of the American Statistical Association* 102.477 (2007). ISSN: 01621459. DOI: 10.1198/016214506000001437.

[22] Axel Brando Guillaumes et al. “Mixture Density Networks for distribution and uncertainty estimation”. In: ().

[23] Chunyang He et al. “Future global urban water scarcity and potential solutions”. In: (). DOI: 10.1038/s41467-021-25026-3. URL: https://doi.org/10.1038/s41467-021-25026-3.

[24] Manuel Herrera et al. “Predictive models for forecasting hourly urban water demand”. In: *Journal of Hydrology* 387.1-2 (June 2010), pp. 141–150. ISSN: 0022-1694. DOI: 10.1016/J.JHYDROL.2010.04.005.

[25] Sepp Hochreiter and Jürgen Schmidhuber. “lstm original paper”. In: (1997).

[26] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods”. In: *Machine Learning* 110.3 (Mar. 2021), pp. 457–506. ISSN: 15730565. DOI: 10.1007/S10994-021-05946-3/FIGURES/17. URL: https://link.springer.com/article/10.1007/s10994-021-05946-3.

[27] Christopher J. Hutton and Zoran Kapelan. “A probabilistic methodology for quantifying, diagnosing and reducing model structural and predictive errors in short term water demand forecasting”. In: *Environmental Modelling and Software* 66 (Apr. 2015), pp. 87–97. ISSN: 13648152. DOI: 10.1016/j.envsoft.2014.12.021.

[28] Rob Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice (3rd ed)*. 2021. URL: https://otexts.com/fpp3/.

[29] Vilde Jensen, Filippo Maria Bianchi, and Stian Normann Anfinsen. “Ensemble Conformalized Quantile Regression for Probabilistic Time Series Forecasting.” In: *IEEE Transactions on Neural Networks and Learning Systems* PP (Nov. 2022). ISSN: 2162-237X. DOI: 10.1109/TNNLS.2022.3217694. URL: https://europepmc.org/article/med/36331651.

[30] Diederik P. Kingma and Jimmy Lei Ba. “Adam: A method for stochastic optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. 2015.

[31] Jens Kley-Holsteg and Florian Ziel. “Probabilistic Multi-Step-Ahead Short-Term Water Demand Forecasting with Lasso”. In: (2020). DOI: 10.1061/(ASCE)WR.1943. URL: https://orcid.org/0000-0002-9911-4096..

[32] Jens Kley-Holsteg et al. “Water Demand Forecasting Based on Online Aggregation for District Meter Areas-Specific Adaption”. In: *Engineering Proceedings 2024, Vol. 69, Page 15* 69.1 (Aug. 2024), p. 15. ISSN: 2673-4591. DOI: 10.3390/ENGPROC2024069015. URL: https://www.mdpi.com/2673-4591/69/1/15/htm%20https://www.mdpi.com/2673-4591/69/1/15.

[33] Daniel Klotz et al. “Uncertainty estimation with deep learning for rainfall-runoff modeling”. In: *HESS* 26.6 (Mar. 2022), pp. 1673–1693. ISSN: 1027-5606. DOI: 10.5194/HESS-26-1673-2022. URL: https://ui.adsabs.harvard.edu/abs/2022HESS...26.1673K/abstract.

[34] Junghwan Lee, Chen Xu, and Yao Xie. “Transformer Conformal Prediction for Time Series”. In: ().

[35] P J de Moel, JQJC Verberk, and J C van Dijk. *Drinking Water: Principles and Practices*. Undefined/Unknown. World Scientific, 2006. ISBN: 981-256-836-0.

[36] Li Mu et al. “Hourly and Daily Urban Water Demand Predictions Using a Long Short-Term Memory Based Model”. In: *Journal of Water Resources Planning and Management* 146.9 (Sept. 2020). ISSN: 0733-9496. DOI: 10.1061/(asce)wr.1943-5452.0001276.

[37] Harrison E. Mutikanga, Saroj K. Sharma, and Kalanithy Vairavamoorthy. “Methods and Tools for Managing Losses in Water Distribution Systems”. In: *Journal of Water Resources Planning and Management* 139.2 (Apr. 2012), pp. 166–174. ISSN: 0733-9496. DOI: 10.1061/(ASCE)WR.1943-5452.0000245. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29WR.1943-5452.0000245%20https://ascelibrary.org/doi/10.1061/%28ASCE%29WR.1943-5452.0000245.

[38] United Nations et al. “World Population Prospects 2019 Highlights”. In: ().

[39] *Optimised control and pipe burst detection by water demand forecasting; case studies in six locations in the Netherlands (dataset)*. URL: https://data.4tu.nl/datasets/22de2cee-0630-4ed4-962d-5dd278d01aaf/1.

[40] Pierre Pinson and Julija Tastu. *Discussion of ”prediction intervals for short-term wind farm generation forecasts” and ”combined nonparametric prediction intervals for wind power generation”*. 2014. DOI: 10.1109/TSTE.2014.2323851.

[41] Irene Pluchinotta et al. “A participatory system dynamics model to investigate sustainable urban water management in Ebbsfleet Garden City”. In: *Sustainable Cities and Society* 67 (Apr. 2021). ISSN: 22106707. DOI: 10.1016/j.scs.2021.102709. URL: https://nottingham-repository.worktribe.com/output/5202243/a-participatory-system-dynamics-model-to-investigate-sustainable-urban-water-management-in-ebbsfleet-garden-city%20https://nottingham-repository.worktribe.com/output/5202243/a-participatory-system-dynamics-model-to-investigate-sustainable-urban-water-management-in-ebbsfleet-garden-city.abstract.

[42] Ana L. Reis et al. *A review of operational control strategies in water supply systems for energy and cost efficiency*. 2023. DOI: 10.1016/j.rser.2022.113140.

[43] Mostafa Rezaali, John Quilty, and Abdolreza Karimi. “Probabilistic urban water demand forecasting using wavelet-based machine learning models”. In: *Journal of Hydrology* 600 (Sept. 2021), p. 126358. ISSN: 0022-1694. DOI: 10.1016/J.JHYDROL.2021.126358.

[44] Herbert Robbins and Sutton Monro. “A Stochastic Approximation Method”. In: *The Annals of Mathematical Statistics* 22.3 (1951). ISSN: 0003-4851. DOI: 10.1214/aoms/1177729586.

[45] Yaniv Romano, Evan Patterson, and Emmanuel J Candès. *Conformalized Quantile Regression*. Tech. rep. URL: https://github.com/yromano/cqr..

[46] Omer Sagi and Lior Rokach. “Ensemble learning: A survey”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (July 2018), e1249. ISSN: 1942-4795. DOI: 10.1002/WIDM.1249. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/widm.1249%20https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1249%20https://wires.onlinelibrary.wiley.com/doi/10.1002/widm.1249.

[47] David Salinas et al. “DeepAR: Probabilistic forecasting with autoregressive recurrent networks”. In: *International Journal of Forecasting* 36.3 (2020). ISSN: 01692070. DOI: 10.1016/j.ijforecast.2019.07.001.

[48] Glenn Shafer and Vladimir Vovk. *A Tutorial on Conformal Prediction*. Tech. rep. 2008, pp. 371–421.

[49] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.

[50] Natasa Tagasovska and David Lopez-Paz. “Frequentist uncertainty estimates for deep learning”. In: ().

[51] Ryan J Tibshirani et al. *Conformal Prediction Under Covariate Shift*. Tech. rep. URL: http://www.alrw.net.

[52] L Uthayakumaran et al. “Impact of Climate Change on Water Demand Making informed planning decisions for demand forecasting”. In: 4 (2019). DOI: 10.21139/wej.2019.012. URL: https://doi.org/10.21139/wej.2019.012.

[53] Ties Van Der Heijden, Peter Palensky, and Edo Abraham. “Probabilistic DAM price forecasting using a combined Quantile Regression Deep Neural Network with less-crossing quantiles”. In: *IECON Proceedings (Industrial Electronics Conference)* 2021-October (Oct. 2021), p. 9589097. DOI: 10.1109/IECON48115.2021.9589097. URL: https://research.tudelft.nl/en/publications/probabilistic-dam-price-forecasting-using-a-combined-quantile-reg.

[54] Stefan Wager, Sida Wang, and Percy Liang. “Dropout Training as Adaptive Regularization”. In: ().

[55] Xiao jun Wang et al. “Adaptation to climate change impacts on water demand”. In: *Mitigation and Adaptation Strategies for Global Change* 21.1 (Jan. 2016), pp. 81–99. ISSN: 15731596. DOI: 10.1007/S11027-014-9571-6/FIGURES/3. URL: https://link-springer-com.tudelft.idm.oclc.org/article/10.1007/s11027-014-9571-6.

[56] Ruofeng Wen et al. “A Multi-Horizon Quantile Recurrent Forecaster”. In: (Nov. 2017). URL: https://arxiv.org/abs/1711.11053v2.

[57] Robert L. Winkler. “A decision-theoretic approach to interval estimation”. In: *Journal of the American Statistical Association* 67.337 (1972). ISSN: 1537274X. DOI: 10.1080/01621459.1972.10481224.

[58] Andrew C Worthington. “Commercial and industrial water demand estimation: Theoretical and methodological guidelines for applied economics research”. In: *Estudios de Economia Aplicada* (2010).

[59] Chen Xu and Yao Xie. “Conformal Prediction for Time Series”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.10 (Oct. 2023), pp. 11575–11587. ISSN: 19393539. DOI: 10.1109/TPAMI.2023.3272339.

[60] Chen Xu and Yao Xie. *Conformal Prediction Interval for Dynamic Time-Series*. Tech. rep. 2021.

[61] Chen Xu and Yao Xie. *Sequential Predictive Conformal Inference for Time Series*. Tech. rep.

[62] Margaux Zaffran et al. *Adaptive Conformal Predictions for Time Series*. Tech. rep.

[63] Dennis Zanutto et al. “A Water Futures Approach on Water Demand Forecasting with Online Ensemble Learning”. In: MDPI AG, Sept. 2024, p. 60. DOI: 10.3390/engproc2024069060.

A. Effect of updating residuals of CQR

In Table B.1 we see the PICP per DMA of the CQR that updates the residuals which are used to re-calibrate the quantiles with and the CQR that does not update the residuals (used for recalibration of the quantiles). The CQR that does not update the residuals online has considerably lower coverages on the testing set. These are comparable on DMA B with the coverages of the MCD, which does not use online updating either. This likely happened because the data on the testing set is very heteroskedastic during the summer months, which is little captured by the validation set on which the MCD is calibrated. The premise of the jointly trained QR model is that it would learn how to generalize this.

B. Data Splits

Table B.1: Comparison of CQR (with and without online update) and QR Across DMAs

	A	B	C	D	E	F	G	H	I	J	Average CG	Max CG
CQR (with online update)	**0.95**	**0.94**	**0.94**	0.94	**0.95**	**0.95**	0.98	**0.95**	**0.95**	**0.94**	**0.01**	**0.03**
MCD	**0.95**	**0.85**	0.92	**0.95**	**0.90**	0.96	**0.96**	**0.89**	**0.95**	**0.94**	**0.03**	**0.10**
CQR (no online update)	0.94	**0.85**	**0.90**	0.93	0.94	0.94	0.98	0.92	0.93	**0.94**	**0.03**	**0.10**
QR	**0.92**	**0.94**	0.93	**0.89**	0.99	**0.90**	**0.99**	0.96	**0.90**	**0.94**	**0.03**	0.06

B.1. Missing Values And Data Splits

Figure B.1: Missing values and data splits (not interpolated)

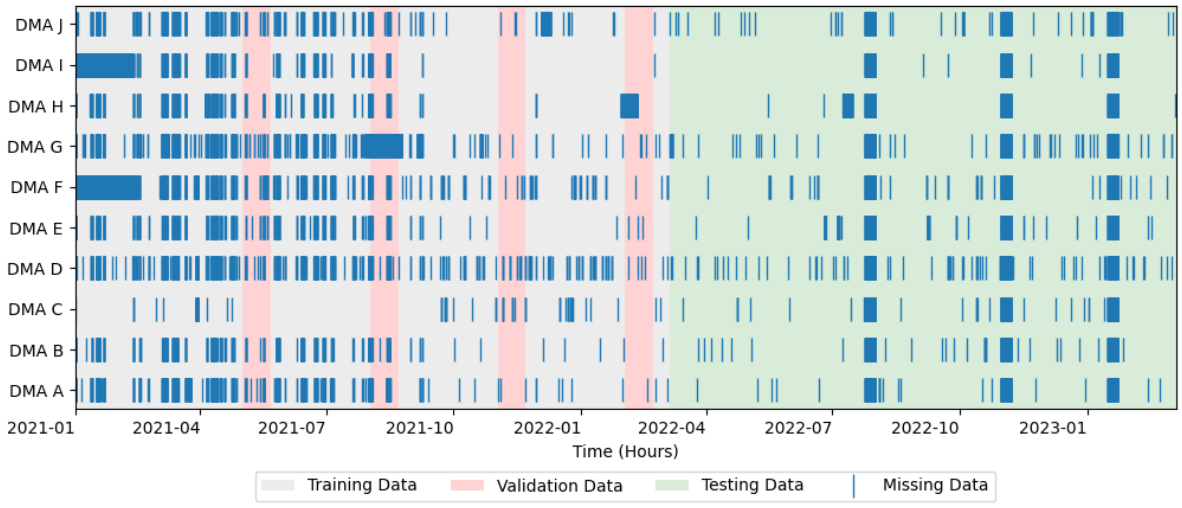
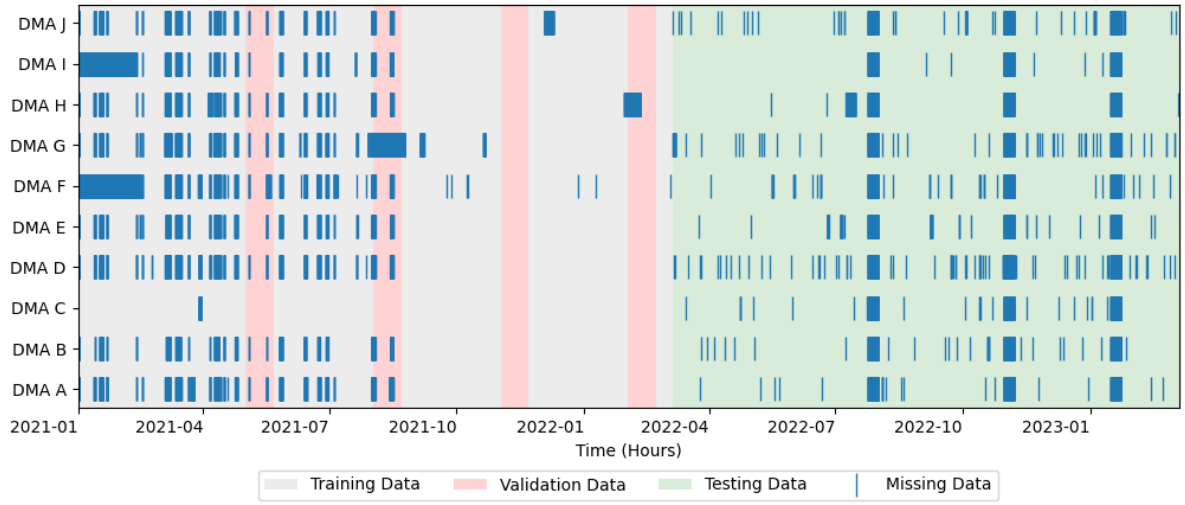


Figure B.2: Missing values and data splits (first year and two months interpolated)



B.2. Tables And Number Of Sequences Per Data Split

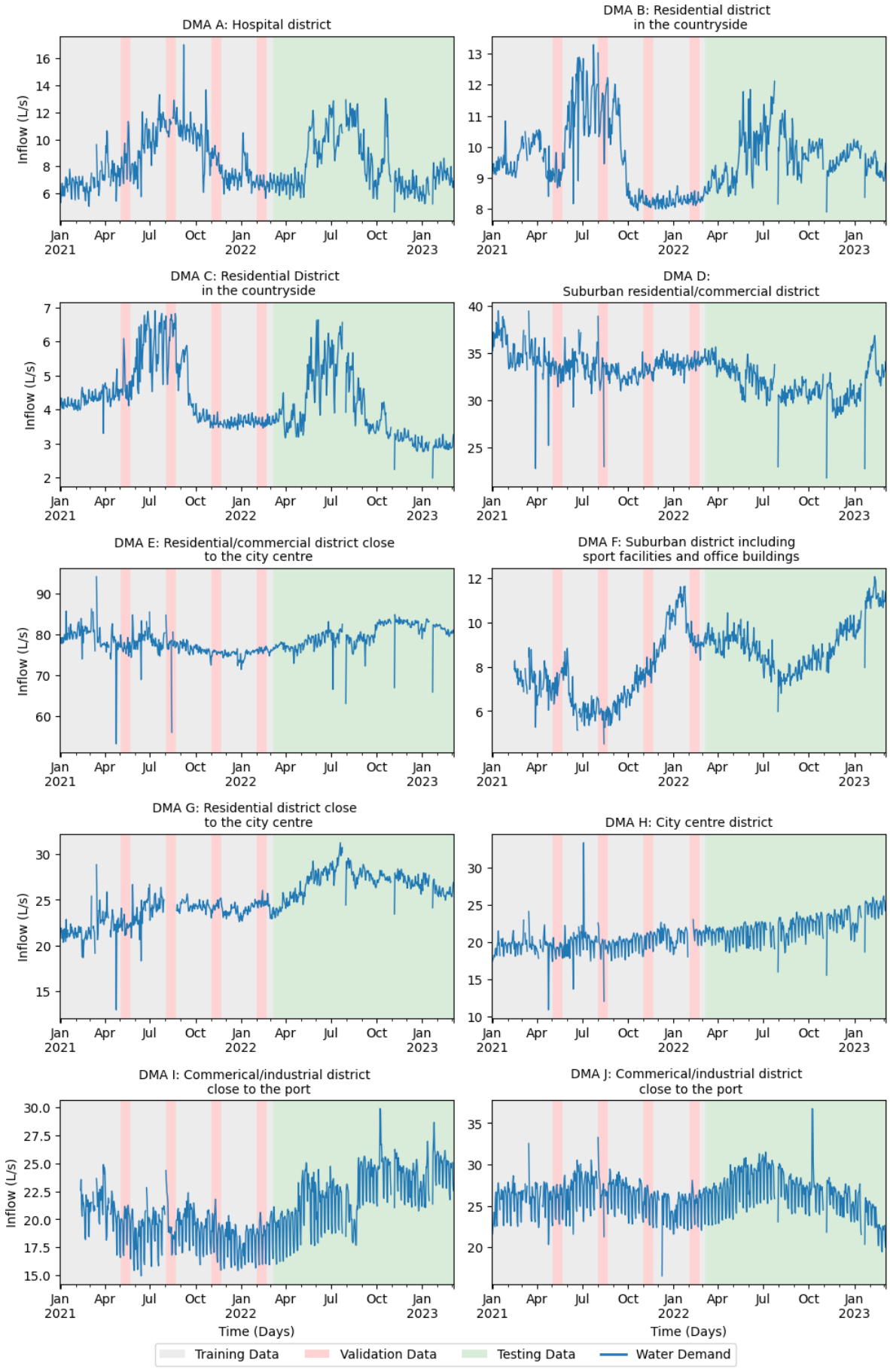
Table B.2: Training and Validation Splits for Water Demand Forecasting

	Training Split 1	Training Split 2	Training Split 3	Training Split 4	Training Split 5	Total Training Sequences	Validation Split 1	Validation Split 2	Validation Split 3	Validation Split 4	Total Validation Sequences	Test Split
DMA A	674	653	1513	1513	96	4449	120	62	313	313	808	5145
DMA B	895	797	1513	1513	96	4814	126	74	313	313	826	4161
DMA C	2471	1513	1513	1513	96	7106	313	313	313	313	1252	4812
DMA D	430	380	1513	1513	96	3932	122	72	313	313	820	1764
DMA E	769	470	1513	1513	96	4361	196	74	313	313	896	4839
DMA F	205	273	1009	1129	60	2676	120	73	313	313	819	4444
DMA G	748	318	1001	1513	96	3676	121	0	313	313	747	2933
DMA H	737	681	1513	1469	96	4496	122	74	313	87	596	6744
DMA I	380	608	1513	1513	96	4110	239	73	313	313	938	6578
DMA J	753	664	1513	1171	96	4197	122	73	313	313	821	3992

One segment of the validation split for DMA G has no data, however this is not expected to give a bias to the result as there is no strong annual seasonality in this DMA see next appendix (B.3).

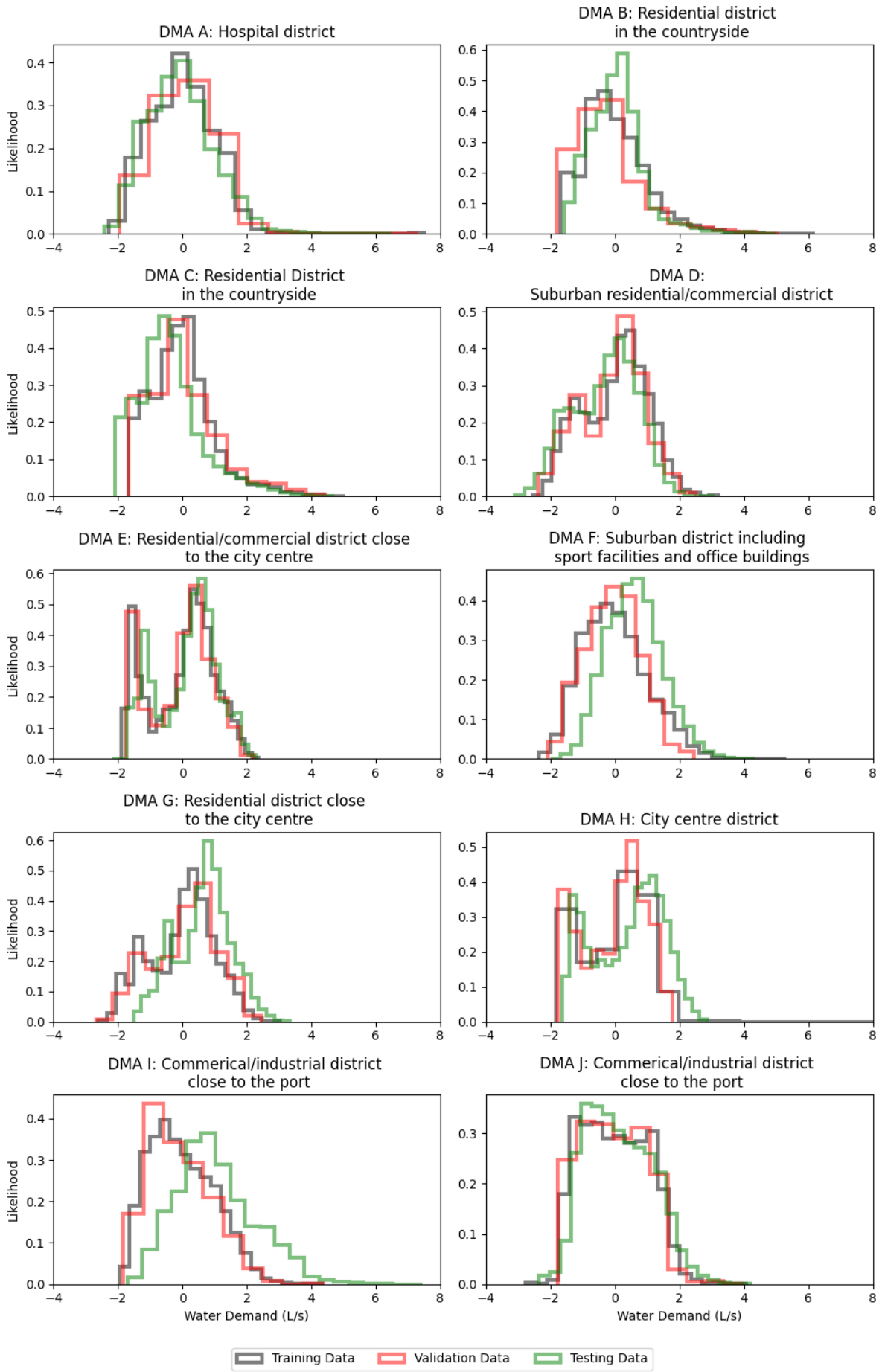
B.3. Time Series Of Each DMA And Data Splits

Figure B.3: Daily inflow for each DMA And training, validation and testing splits (peaks and troughs are result of missing values in data aggregation))



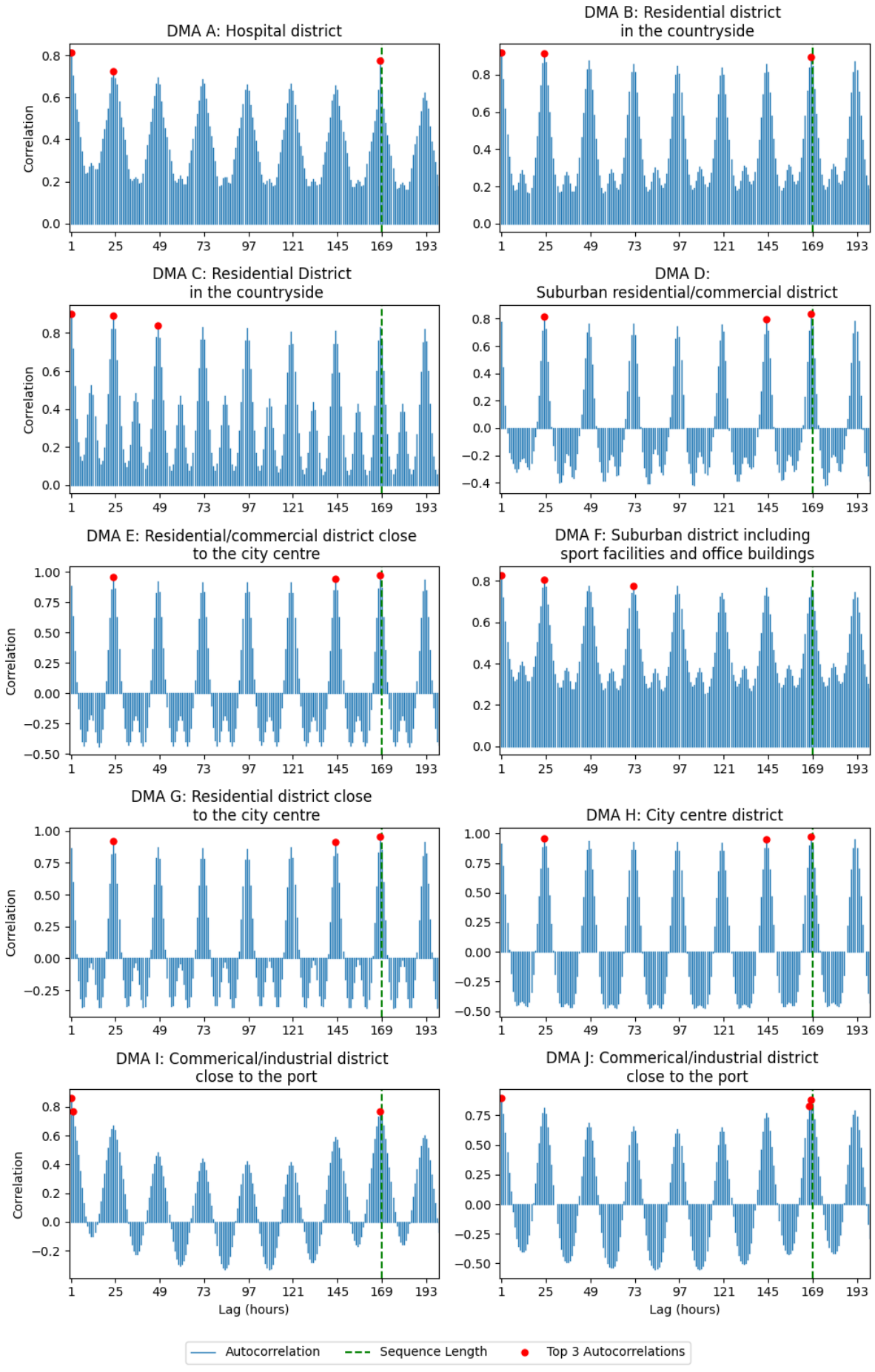
B.4. Distributions Of Data Splits

Figure B.4: Distribution of each data split



B.4.1. Auto-Correlations Per DMA

Figure B.5: Auto-Correlations of each data split



C. Water Demand Pattern

Figure C.1: Pattern Water Demand)

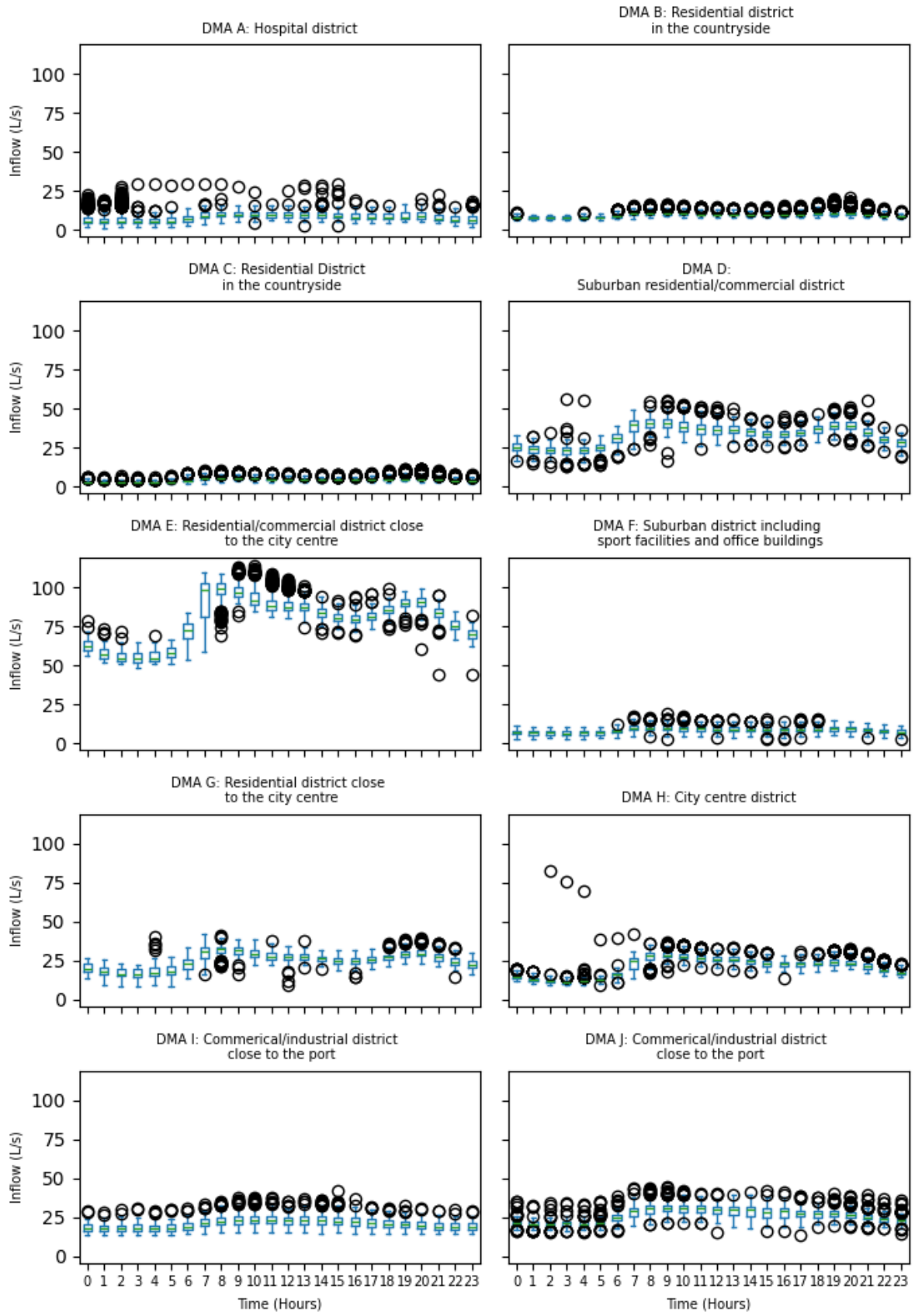


Figure C.2: Pattern Water Demand divided between weekend and weekday (DMA A to E)

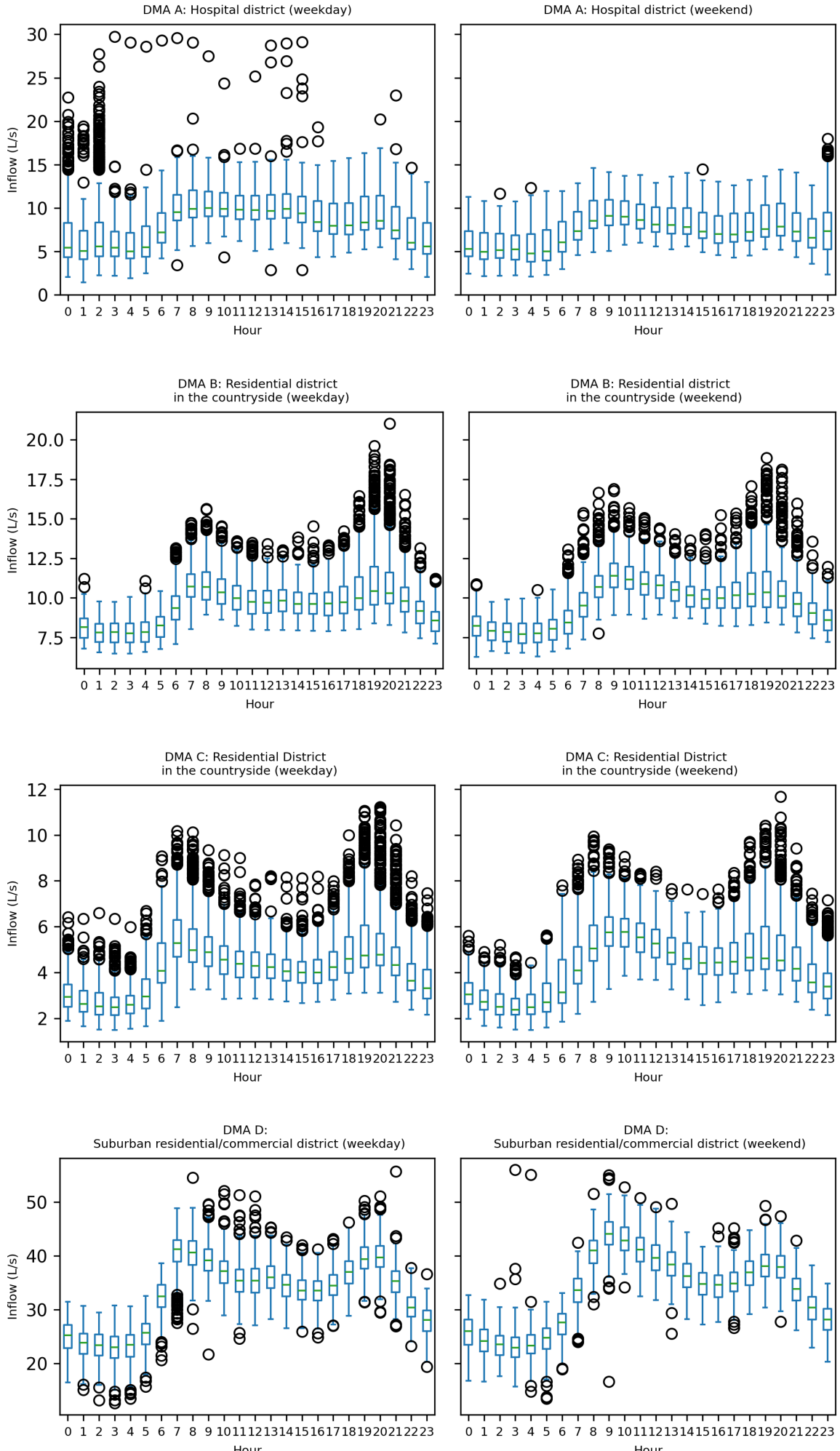
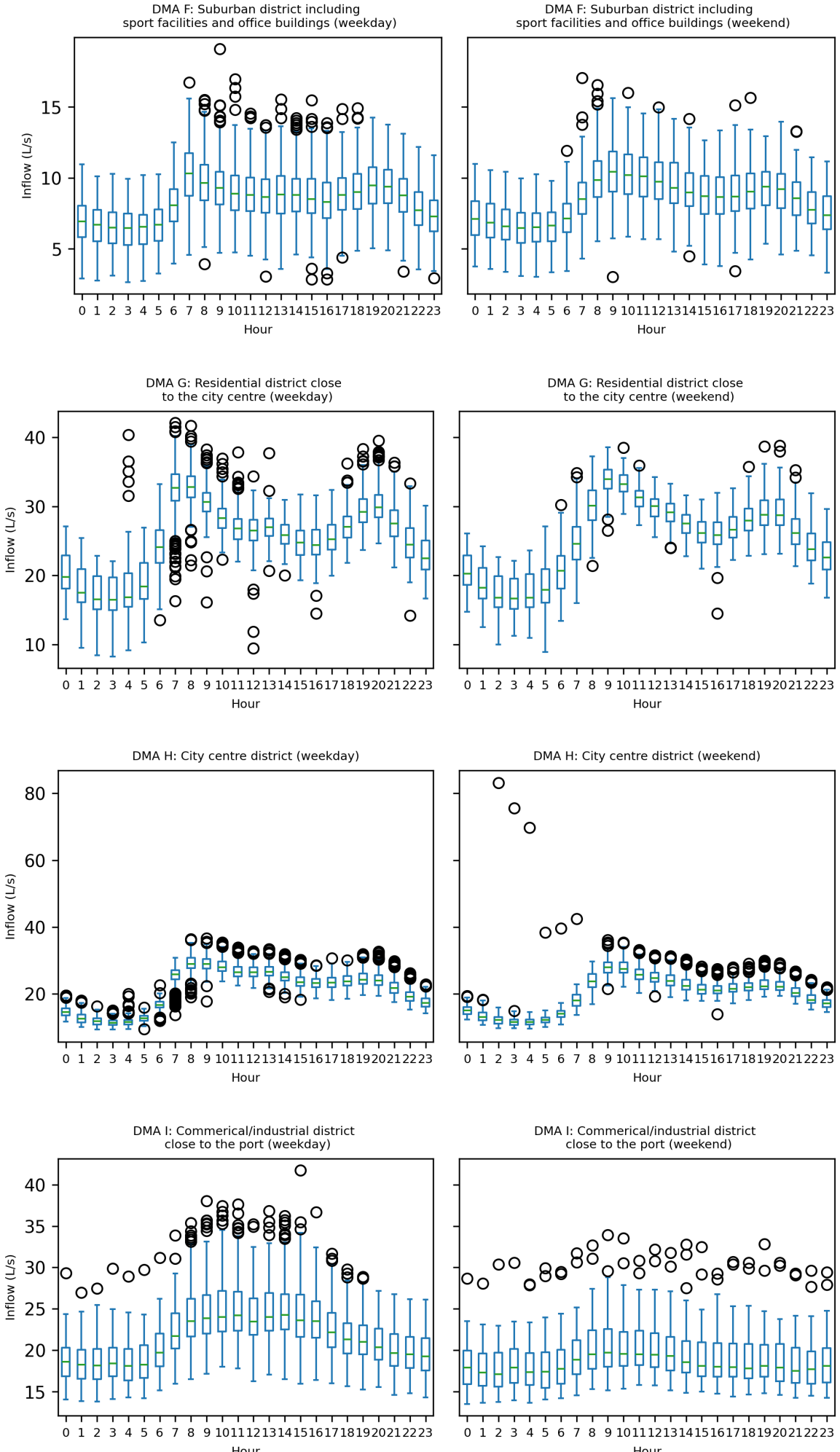


Figure C.3: Pattern Water Demand divided between weekend and weekday (DMA F to J)



D. Hyperparameter Search Rsults

D.1. Hyperparameters Configuration Results Deterministic Prediction Models

Table D.1: Configuration Parameters for models trained per DMA (MLP model on CPU, Linear and LSTM on Nvidia V100 GPU)

Linear Model	A	B	C	D	E	F	G	H	I	J
Dropout Rate	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Hidden Size	128	128	128	128	128	128	128	128	64	128
Learning Rate	4.03×10^{-4}	1.62×10^{-3}	3.64×10^{-4}	2.80×10^{-4}	7.98×10^{-4}	1.42×10^{-3}	9.30×10^{-4}	1.41×10^{-3}	9.25×10^{-4}	3.84×10^{-4}
Number Epochs	250	250	250	250	250	250	250	250	250	250
Batch Size	256	256	256	256	256	256	256	256	256	256
Number of Parameters	24,728	24,728	24,728	24,728	24,728	24,728	24,728	24,728	12,376	24,728
Train Time (min)	0.256	0.273	0.380	0.229	0.256	0.169	0.208	0.275	0.236	0.248
MLP	A	B	C	D	E	F	G	H	I	J
Dropout Rate	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Hidden Size	128	128	128	128	128	128	128	128	128	128
Hidden Layers	0	3	2	2	3	0	3	3	2	3
Learning Rate	2.53×10^{-4}	2.37×10^{-3}	2.04×10^{-3}	2.59×10^{-4}	2.77×10^{-3}	3.75×10^{-4}	1.86×10^{-3}	3.20×10^{-3}	6.52×10^{-4}	2.26×10^{-3}
Number Epochs	250	250	250	250	250	250	250	250	250	250
Batch Size	256	256	256	256	256	256	256	256	256	256
Number of Parameters	24,728	74,264	57,752	57,752	74,264	24,728	74,264	74,264	57,752	74,264
Train Time (min)	0.244	0.305	0.410	0.276	0.288	0.156	0.257	0.288	0.266	0.280
LSTM	A	B	C	D	E	F	G	H	I	J
Dropout Rate	0.1	0.1	0.1	0.15	0.1	0.1	0.2	0.15	0.2	0.15
Hidden Size	128	128	128	128	128	128	128	128	128	128
Learning Rate	7.69×10^{-3}	8.12×10^{-3}	9.93×10^{-3}	1.97×10^{-3}	7.81×10^{-3}	4.76×10^{-3}	7.34×10^{-3}	4.47×10^{-3}	3.50×10^{-3}	5.42×10^{-3}
Number Epochs	250	250	250	250	250	250	250	250	250	250
Batch Size	256	256	256	256	256	256	256	256	256	256
Number of Parameters	70,168	70,168	70,168	70,168	70,168	70,168	70,168	70,168	70,168	70,168
Train Time (min)	0.598	0.631	0.915	0.527	0.605	0.380	0.495	0.588	0.565	0.599

Table D.2: Configuration parameters found in hyperparameter search one model for all DMAs. (Trained on Nvidia V100 GPU)

Model Name	**Dropout Rate**	**Hidden Size**	**Hidden Layers**	**Learning Rate**	**Number of Epochs**	**Batch Size**	**Number of Parameters**	**Train Time (min)**	**Extra Notes**
Linear	0.1	256	N/A	8.65×10^{-5}	250	256	49,432	2.24	
MLP	0.1	256	2	1.17×10^{-4}	250	256	181,016	2.83	
LSTM	0.1	128	1	1.06×10^{-3}	250	256	70,168	5.88	
MLP + One-hot DMA indicator	0.1	256	2	3.19×10^{-4}	250	256	183,576	3.24	
LSTM + One-hot DMA indicator	0.1	128	2	9.99×10^{-3}	250	256	204,352	10.19	One-hot Output Size: 4

D.1.1. Hyperparameter Configuration Results Probabilistic Forecasts

Table D.3: Configuration Parameters for Probabilistic Models, trained on NVIDIA V100 GPU

Model Name	**Dropout Rate**	**Hidden Size**	**Hidden Layers**	**Learning Rate**	**Number of Epochs**	**Batch Size**	**Number of Parameters**	**Train Time (min)**	**Number of Gaussians**
QR	0.15	256	1	2.53×10^{-3}	250	256	127560	2.85	N/A
MDN	0.2	256	1	5.54×10^{-4}	250	256	183072	3.41	4

Table D.4: Dropout Rates for DMAs A to J for MCD

DMA	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
Dropout Rate	0.5281	0.2891	0.3594	0.4719	0.1063	0.5844	0.2328	0.1766	0.5000	0.4156

D.1.2. Benchmark Model

The benchmark model is optimized over the first 14 months of data (training and validation set together). Here the error is minimized by finding the lowest MAE by taking an average of 1 to 4 weeks ago. 1 week ago means the value of 1 week ago.

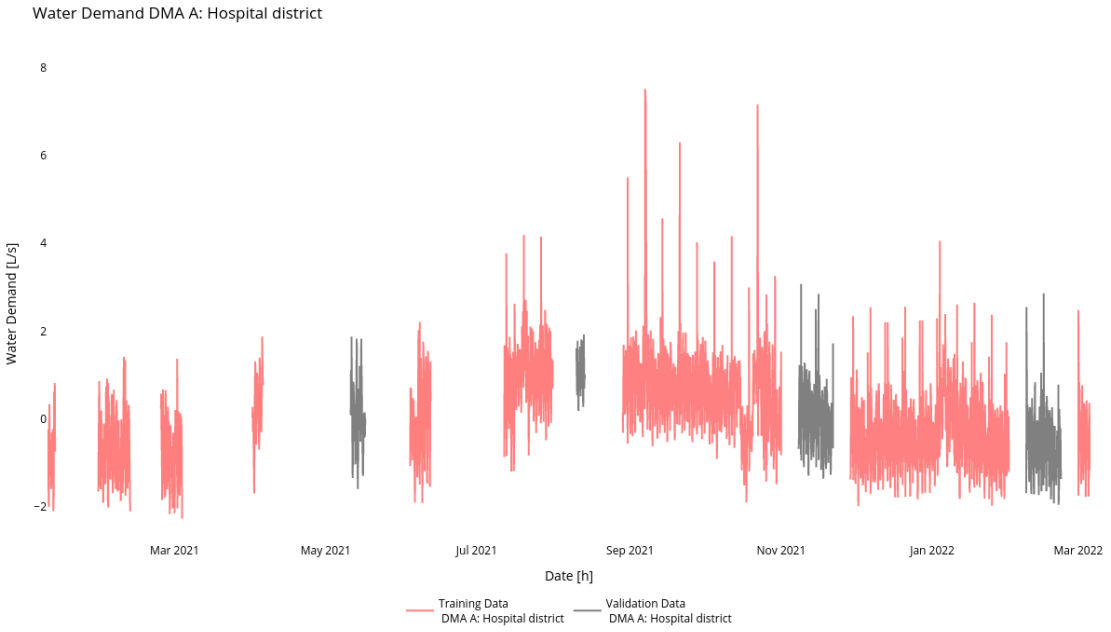
Table D.5: Bold values are the best score. The row number are the number of values the moving average is made with of the benchmark model.

	A	B	C	D	E	F	G	H	I	J
N										
0	1.24	**0.58**	0.45	2.63	1.95	1.01	1.27	0.99	1.56	1.58
1	1.17	0.59	0.44	2.38	**1.94**	0.95	1.22	0.97	1.43	1.41
2	1.13	0.60	**0.42**	2.30	2.01	**0.95**	**1.21**	0.96	1.39	1.36
3	**1.12**	0.62	0.44	**2.27**	2.09	0.97	1.22	**0.95**	**1.35**	**1.33**

E. Water Demand Data

Note that these data are composed from the Y data used in the loss, the 168 values of model input are not necessarily in the plots below

Figure E.1: Water Demand for DMA A (Training, Validation, and Test Data)



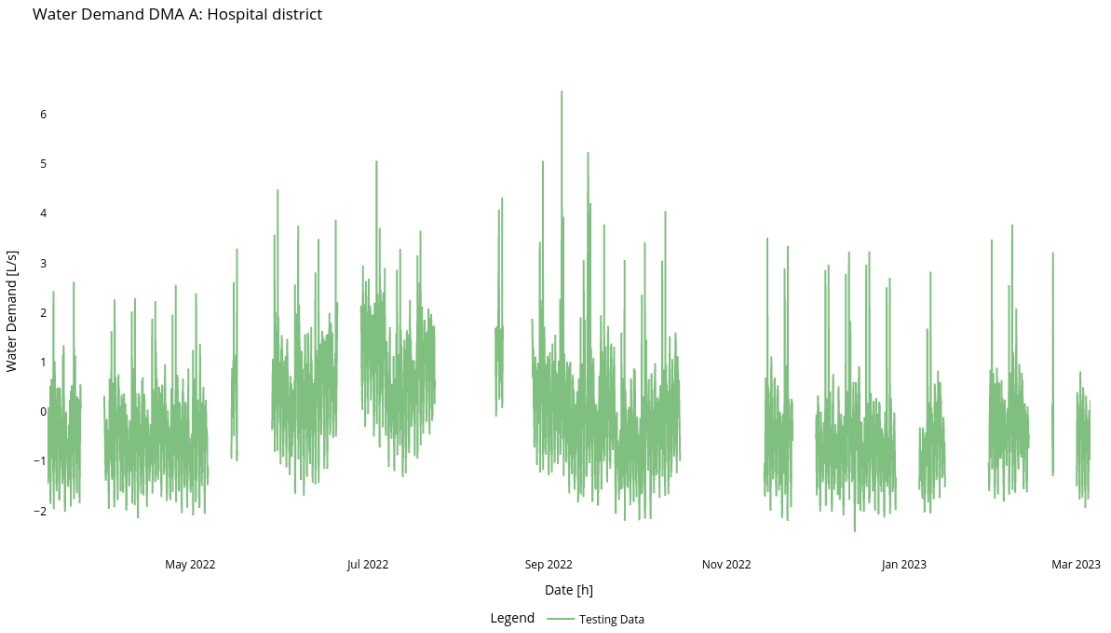


Figure E.2: Water Demand for DMA B (Training, Validation, and Test Data)

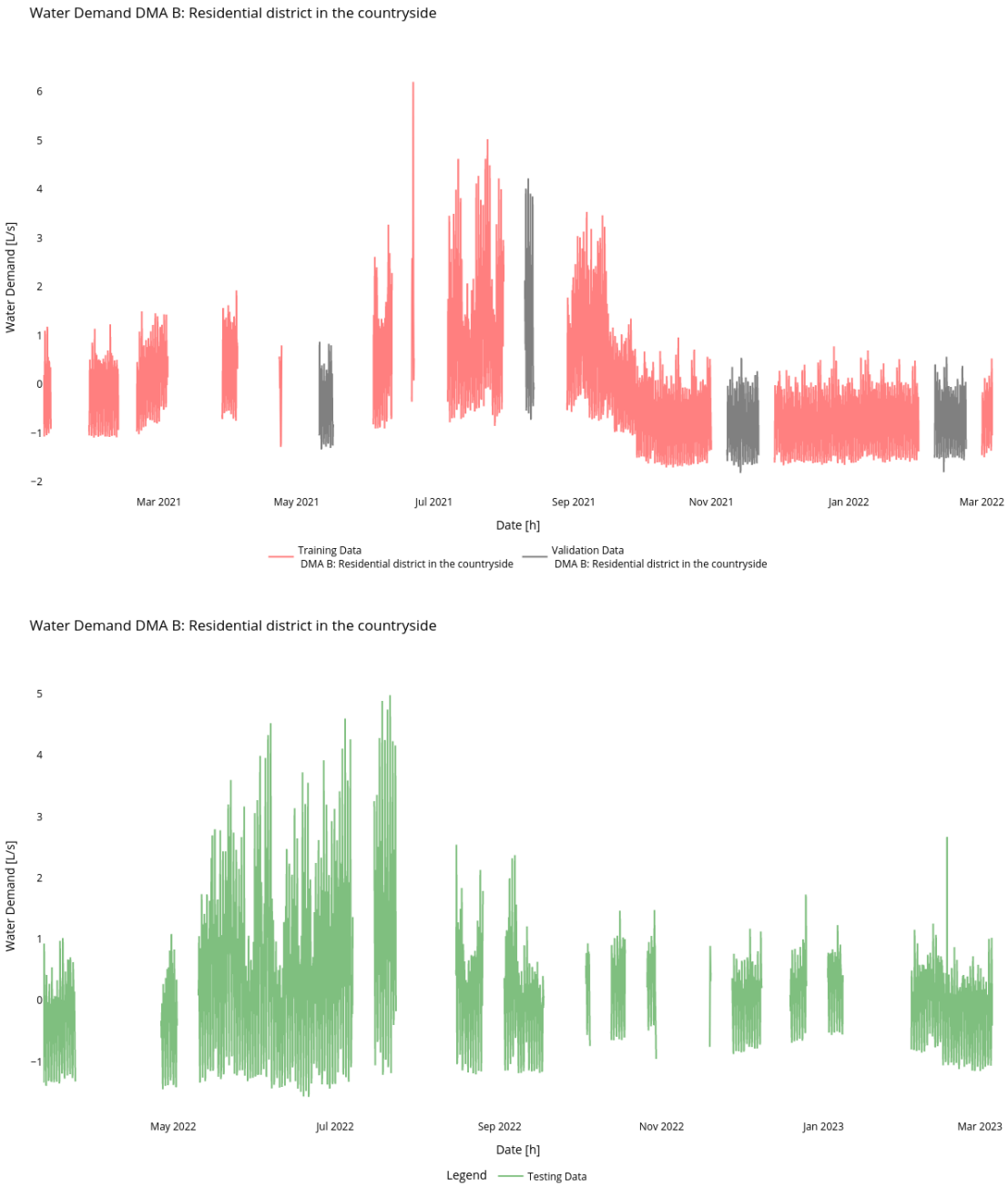


Figure E.3: Water Demand for DMA C (Training, Validation, and Test Data)

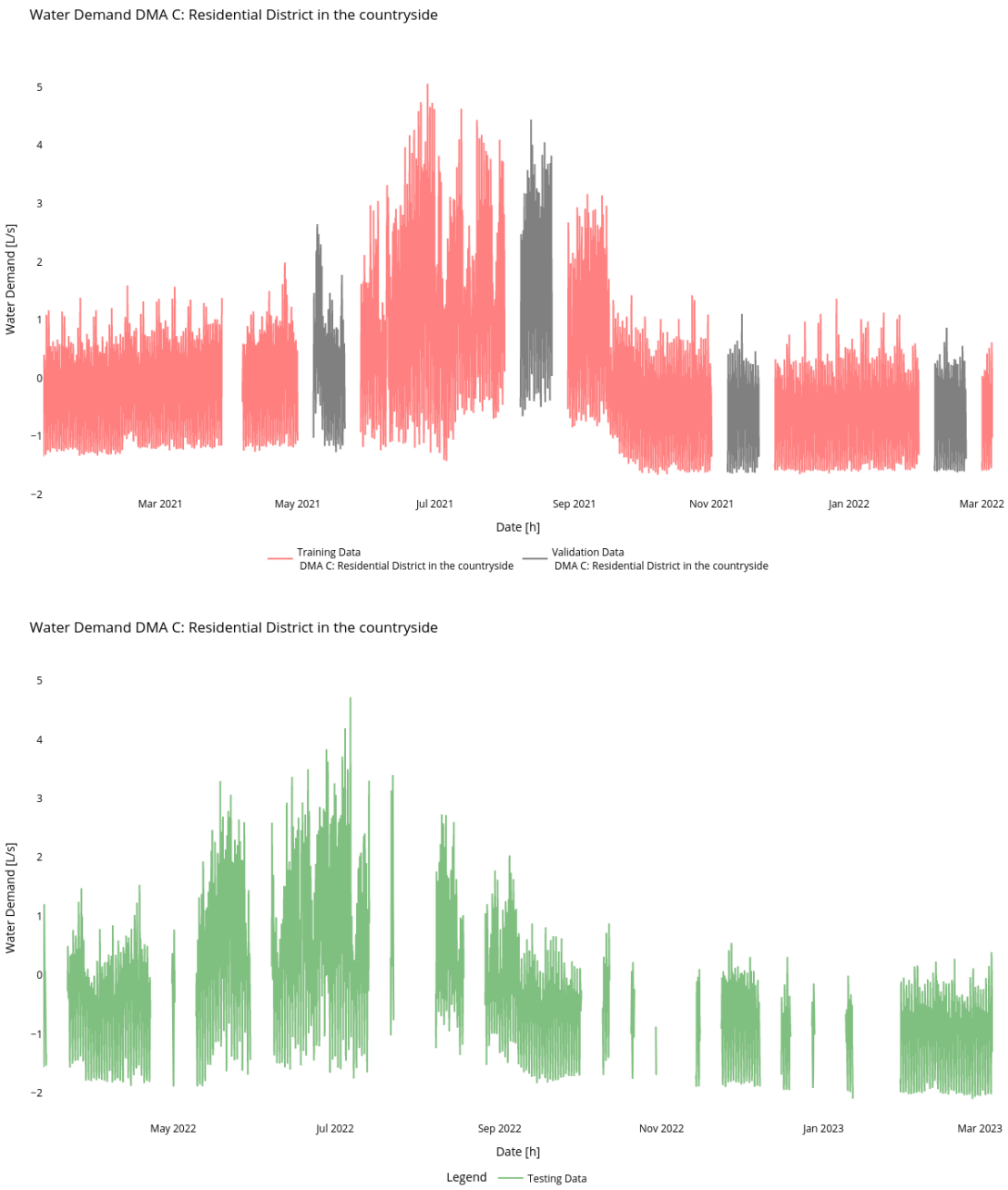


Figure E.4: Water Demand for DMA D (Training, Validation, and Test Data)

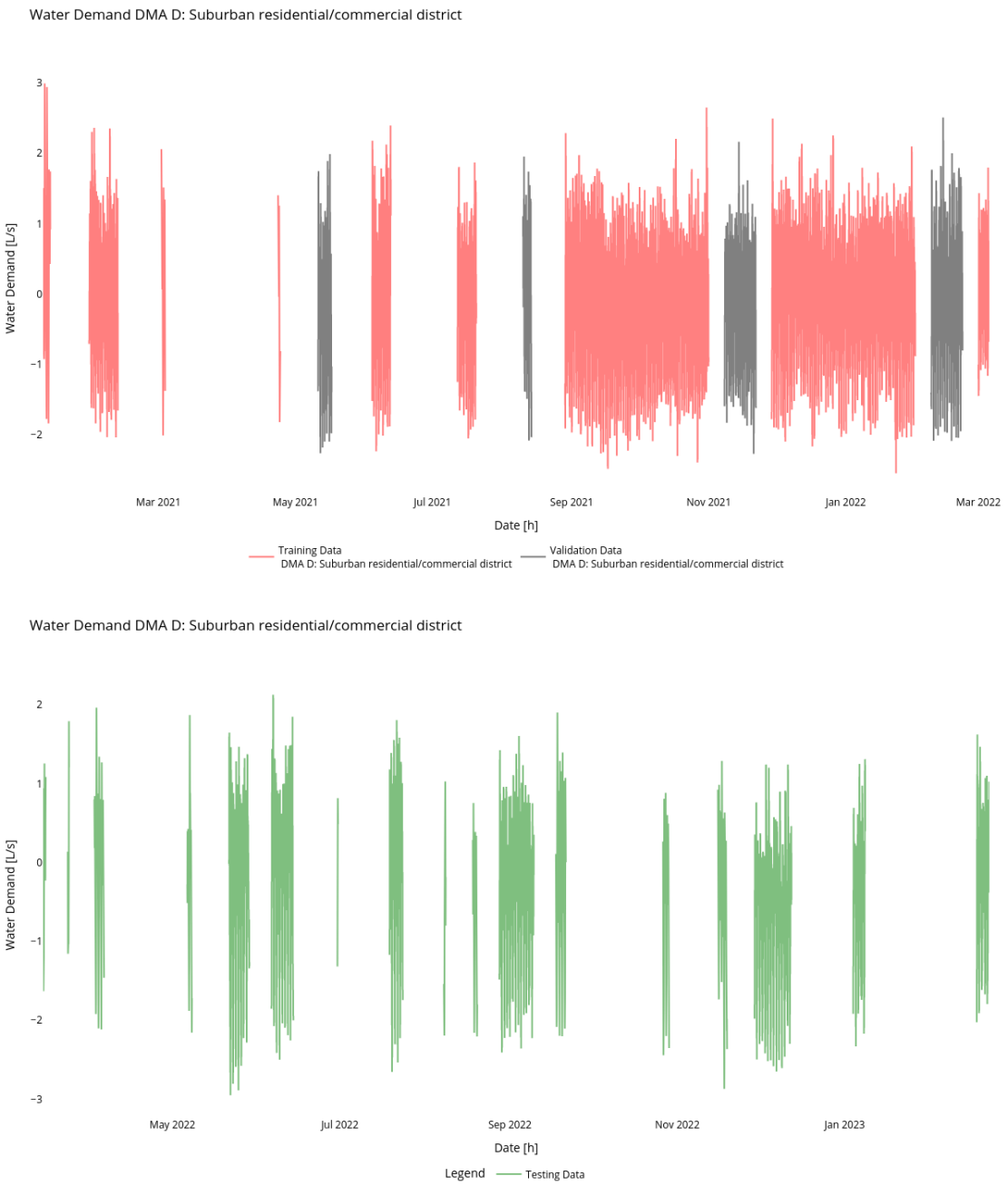


Figure E.5: Water Demand for DMA E (Training, Validation, and Test Data)

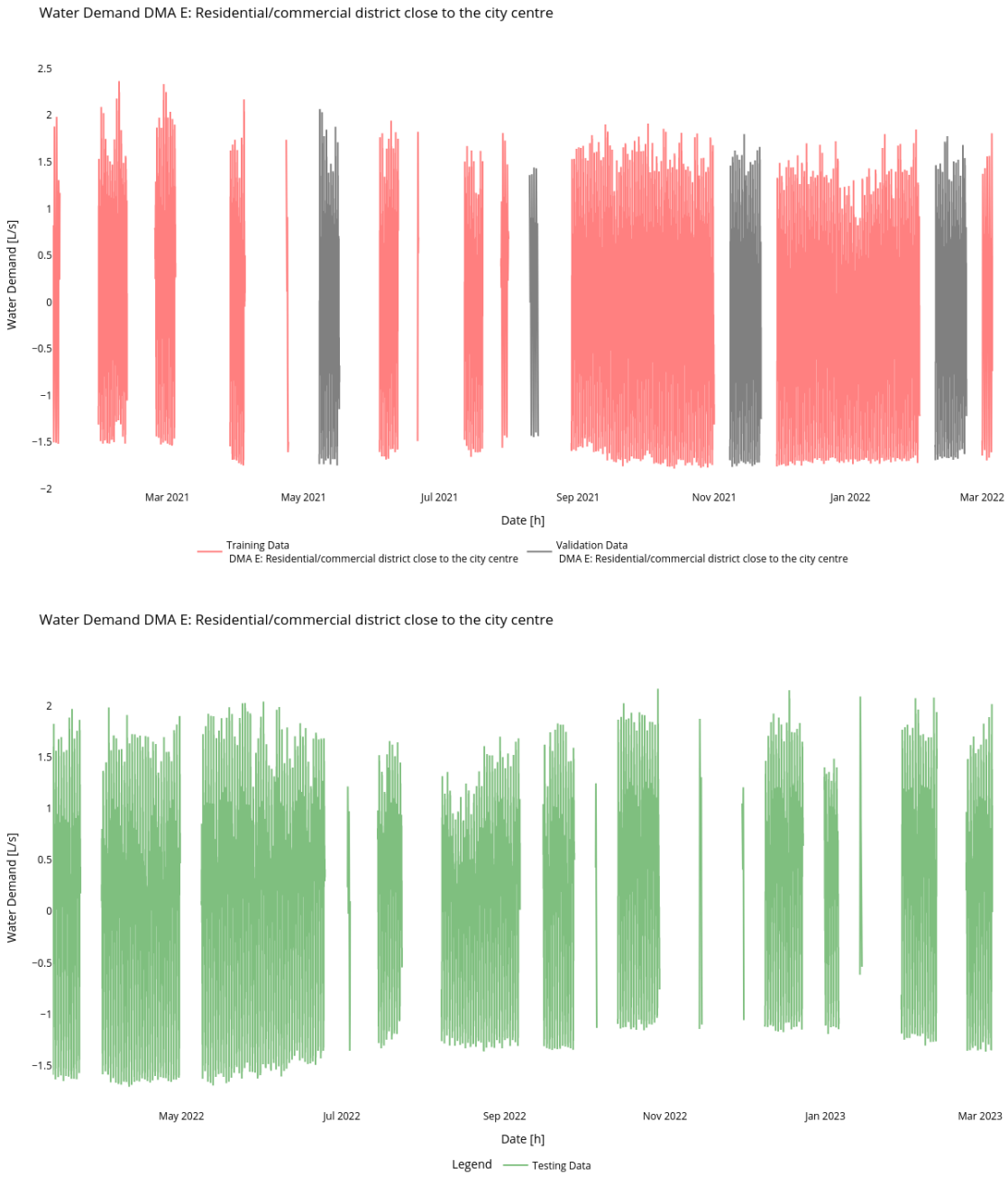


Figure E.6: Water Demand for DMA F (Training, Validation, and Test Data)

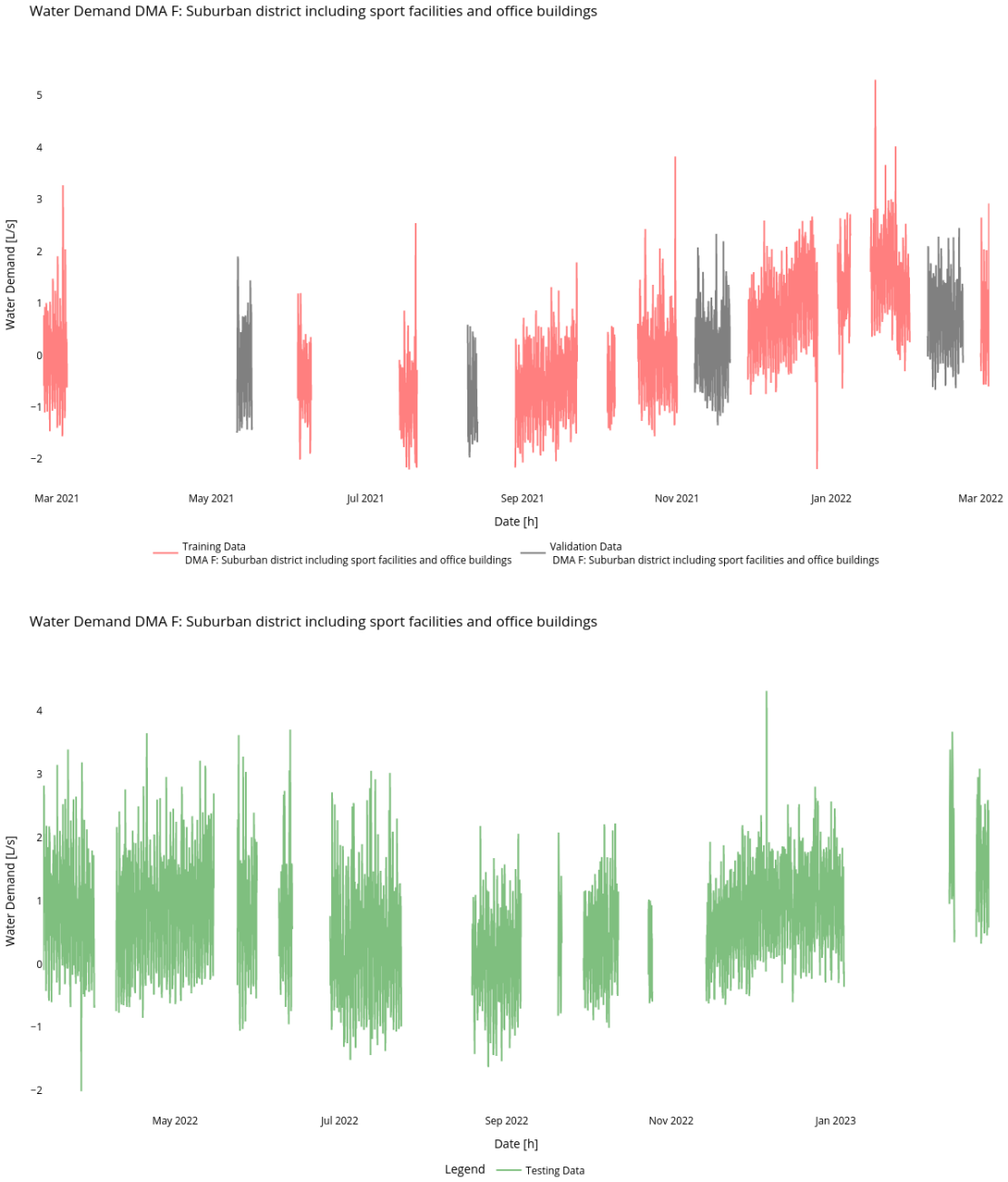


Figure E.7: Water Demand for DMA G (Training, Validation, and Test Data)

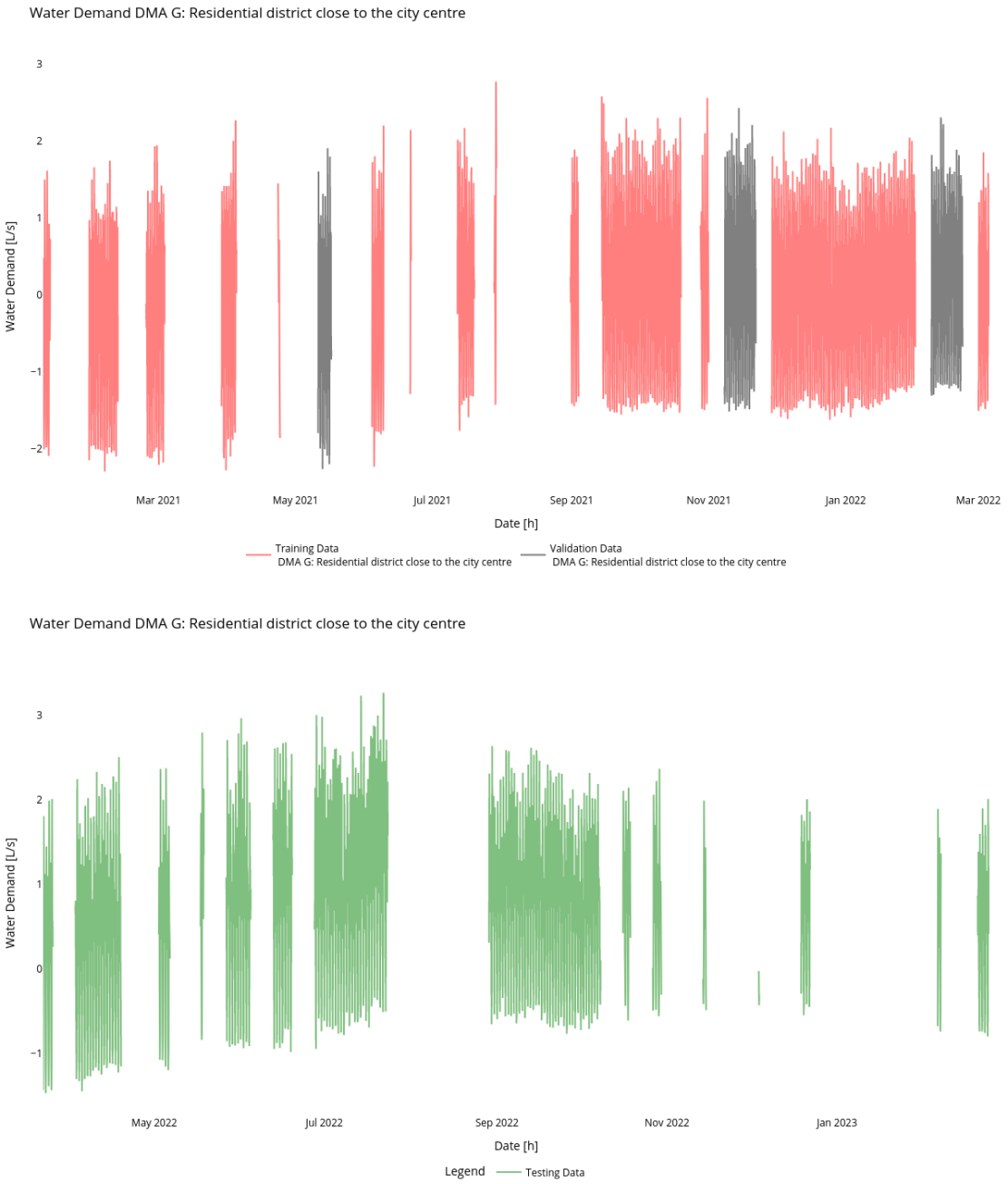


Figure E.8: Water Demand for DMA H (Training, Validation, and Test Data)

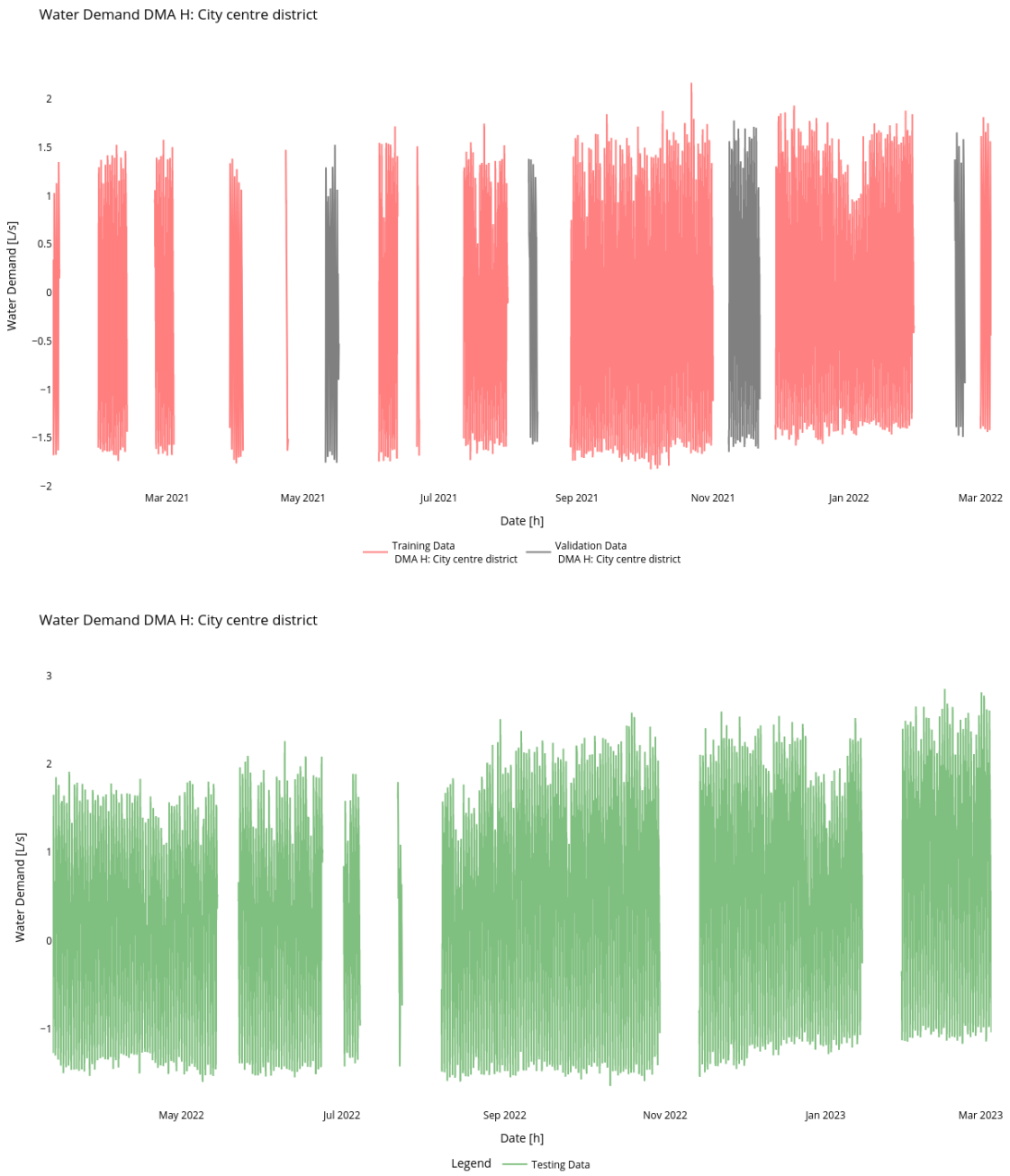


Figure E.9: Water Demand for DMA I (Training, Validation, and Test Data)

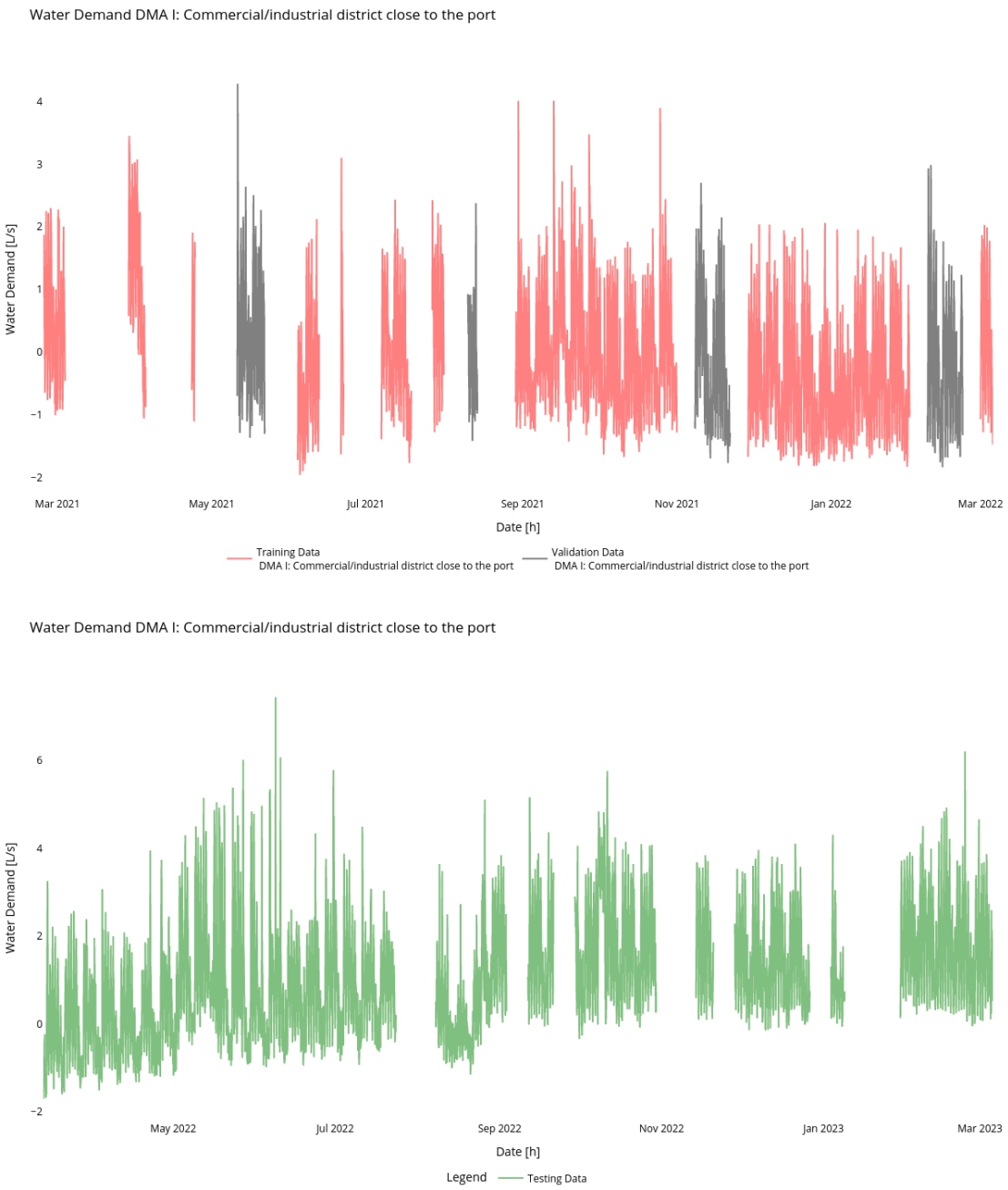
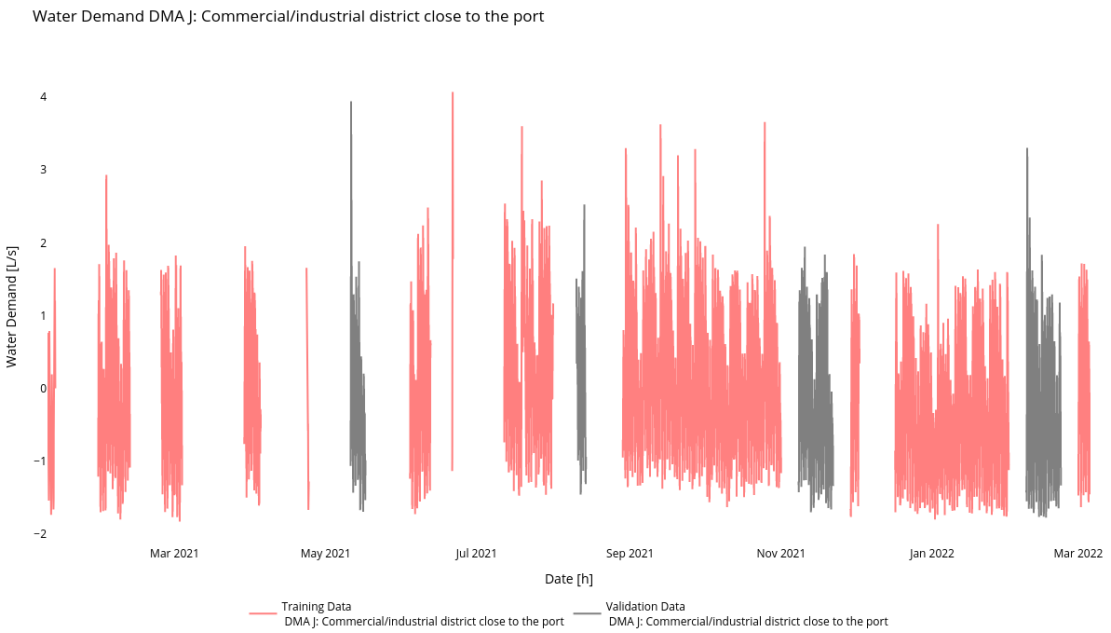
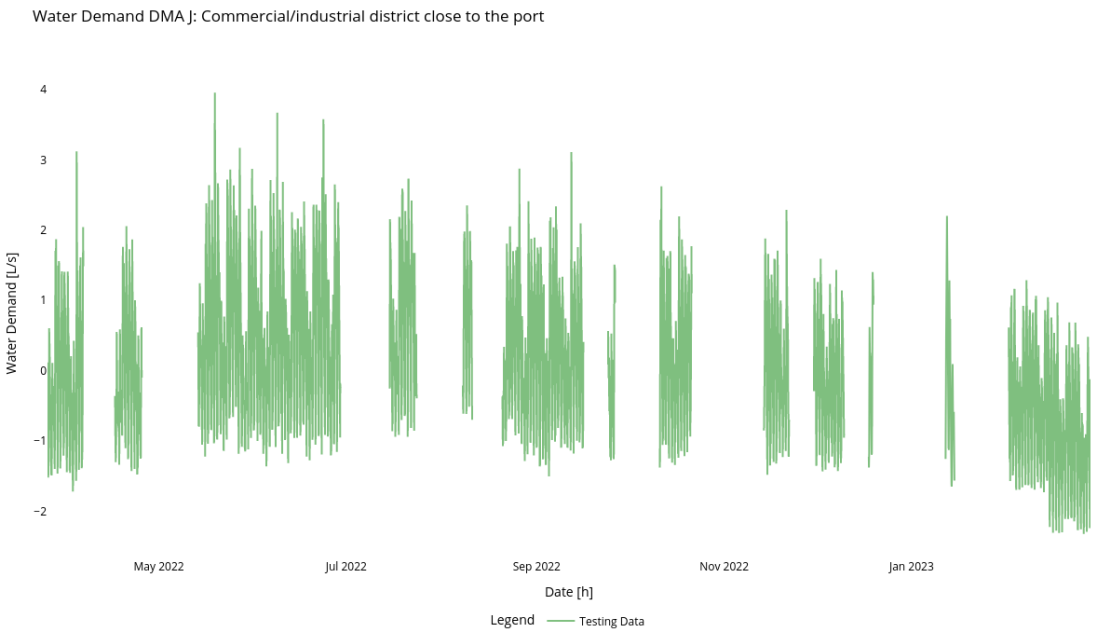


Figure E.10: Water Demand for DMA J (Training, Validation, and Test Data)





F. All Results Point Forecasts

F.1. Tables Average Point Forecasts from Deterministic Models

Table F.1: MAPE [%] point forecasts of all deterministic models on test set

	A	B	C	D	E	F	G	H	I	J	Average
MLP trained on all DMAs	13.52	**4.71**	9.34	**6.96**	**1.81**	8.86	3.76	4.52	6.11	4.67	**6.43**
MLP DMA Indicator trained on all DMAs	**13.38**	4.90	9.70	7.04	1.89	**8.80**	4.01	**4.50**	6.19	**4.64**	6.51
Linear Model trained per DMA	13.98	4.87	**9.30**	7.54	2.33	8.96	3.91	4.95	**6.08**	5.20	6.71
Linear Model trained on all DMAs	15.15	4.81	9.37	7.30	2.25	8.92	**3.53**	4.69	6.13	5.26	6.74
LSTM trained on all DMAs	14.88	5.35	11.31	7.37	2.32	9.53	5.57	5.57	7.62	5.50	7.50
LSTM DMA Indicator trained on all DMAs	13.97	5.63	12.52	7.73	2.19	9.55	6.70	5.64	**7.74**	5.14	7.68
MLP trained per DMA	**13.38**	5.95	11.67	7.61	**2.97**	8.99	8.27	6.85	6.67	4.94	7.73
Benchmark Model	**18.49**	**6.65**	12.64	8.02	2.06	**9.71**	4.00	4.80	7.08	5.54	7.90
LSTM trained per DMA	14.99	6.50	**15.26**	**8.88**	2.67	9.56	**10.08**	**7.32**	7.05	**5.94**	**8.83**

Table F.2: RMAE [-] point forecasts of deterministic models on test set

	A	B	C	D	E	F	G	H	I	J	Average	median	Min	Max
MLP trained on all DMAs	0.79	**0.70**	**0.71**	**0.87**	**0.89**	0.91	0.93	0.96	0.88	0.87	**0.85**	**0.88**	**0.70**	**0.96**
MLP DMA Indicator trained on all DMAs	0.80	0.73	0.74	0.88	0.91	**0.90**	1.00	**0.95**	0.89	**0.86**	0.87	0.89	0.73	1.00
Linear Model trained on all DMAs	0.84	0.72	0.73	0.92	1.10	0.92	**0.89**	0.98	**0.87**	0.98	0.90	0.91	0.72	1.10
Linear Model trained per DMA	0.80	0.72	0.72	0.94	1.12	0.93	0.98	1.03	**0.87**	0.96	0.91	0.94	0.72	1.12
LSTM trained on all DMAs	**0.87**	0.81	0.84	0.93	1.13	**0.99**	1.39	1.18	1.11	1.03	1.03	1.01	0.81	1.39
LSTM DMA Indicator trained on all DMAs	0.84	0.83	0.92	0.96	1.05	**0.99**	1.62	1.22	**1.13**	0.96	1.05	0.98	0.83	1.62
MLP trained per DMA	**0.78**	0.87	0.85	0.94	**1.36**	0.92	2.05	1.48	0.96	0.92	1.11	0.93	0.78	2.05
LSTM trained per DMA	0.86	**0.96**	**1.07**	**1.09**	1.27	0.98	**2.57**	**1.63**	1.02	**1.13**	**1.26**	**1.08**	**0.86**	**2.57**

Table F.3: MAE [L/s] point forecasts of all deterministic models on test set

	A	B	C	D	E	F	G	H	I	J
MLP trained on all DMAs	1.01	**0.49**	**0.38**	**2.02**	**1.48**	**0.78**	1.04	1.02	1.43	1.24
MLP DMA Indicator trained on all DMAs	1.01	0.50	0.39	2.04	1.52	**0.78**	1.11	**1.00**	1.45	**1.23**
Linear Model trained per DMA	1.02	0.50	**0.38**	2.20	1.87	0.80	1.09	1.08	1.41	1.37
Linear Model trained on all DMAs	1.07	0.50	0.39	2.14	1.84	0.79	**0.99**	1.03	**1.40**	1.40
LSTM trained on all DMAs	1.10	0.56	0.44	2.15	1.89	0.85	1.55	1.24	1.79	1.48
LSTM DMA Indicator trained on all DMAs	1.06	0.58	0.49	2.24	1.75	0.85	1.80	1.29	**1.83**	1.37
MLP trained per DMA	**0.99**	0.60	0.45	2.19	**2.27**	0.79	2.29	1.57	1.56	1.31
Benchmark Model	**1.27**	**0.69**	0.53	2.33	1.67	**0.86**	1.11	1.06	1.62	1.43
LSTM trained per DMA	1.09	0.66	**0.56**	**2.54**	2.12	0.84	**2.86**	**1.73**	1.65	**1.61**

Table F.4: MAPE [%] point forecasts of all deterministic models on validation set

	A	B	C	D	E	F	G	H	I	J	Average
MLP trained on all DMAs	11.41	2.78	**6.24**	5.66	1.52	9.12	3.31	3.07	5.69	4.17	**5.30**
MLP DMA Indicator trained on all DMAs	11.54	**2.71**	6.29	**5.65**	1.54	**9.04**	**3.26**	3.11	5.67	4.21	**5.30**
MLP trained per DMA	**10.76**	2.89	6.57	5.76	1.72	9.26	3.40	3.23	5.74	4.23	5.36
LSTM trained per DMA	11.87	2.74	6.94	5.95	**1.50**	9.09	3.76	**2.99**	**5.55**	**4.07**	5.44
LSTM trained on all DMAs	12.27	3.01	6.87	6.04	1.86	9.29	3.45	3.32	6.03	4.19	5.63
LSTM DMA Indicator trained on all DMAs	13.56	2.78	6.99	5.71	1.80	9.38	3.33	3.17	5.81	4.32	5.68
Linear Model trained per DMA	11.17	3.44	7.19	6.00	2.33	9.41	4.03	4.08	6.40	4.74	5.88
Linear Model trained on all DMAs	12.54	3.29	7.12	5.76	**2.36**	9.24	3.98	4.00	**6.55**	**4.83**	5.97
Benchmark Model	**14.44**	**3.73**	**7.88**	**6.61**	1.96	**11.09**	**4.41**	**5.60**	6.46	4.81	**6.70**

Table F.5: RMAE [-] point forecast of all deterministic models on validation set

	A	B	C	D	E	F	G	H	I	J	Average	median	Min	Max
MLP DMA Indicator trained on all DMAs	0.81	**0.68**	**0.79**	**0.85**	0.79	**0.81**	**0.73**	0.59	0.88	0.88	**0.78**	0.80	0.59	**0.88**
MLP trained on all DMAs	0.80	0.70	**0.79**	**0.85**	0.79	0.82	0.75	0.58	0.89	0.87	**0.78**	**0.79**	0.58	0.89
MLP trained per DMA	**0.75**	0.73	0.82	0.87	0.87	0.83	0.76	0.60	0.90	0.88	0.80	0.83	0.60	0.90
LSTM trained per DMA	0.85	**0.68**	0.86	0.90	**0.77**	0.82	0.86	**0.55**	**0.87**	**0.84**	0.80	0.85	**0.55**	0.90
LSTM DMA Indicator trained on all DMAs	**0.96**	0.70	0.89	**0.85**	0.93	**0.84**	0.75	0.59	0.90	0.90	0.83	0.87	0.59	0.96
LSTM trained on all DMAs	0.87	0.76	0.87	0.91	0.96	0.83	0.79	0.61	0.95	0.87	0.84	0.87	0.61	0.96
Linear Model trained per DMA	0.77	**0.87**	0.89	**0.92**	1.18	**0.84**	**0.92**	**0.76**	0.99	0.99	0.91	**0.90**	**0.76**	1.18
Linear Model trained on all DMAs	0.84	0.85	**0.90**	0.88	**1.22**	0.82	0.91	0.75	**1.01**	**1.01**	**0.92**	0.89	0.75	**1.22**

Table F.6: MAE [L/s] point forecast of all deterministic models on validation set

	A	B	C	D	E	F	G	H	I	J
MLP trained on all DMAs	0.86	**0.25**	**0.31**	1.81	1.19	0.72	0.80	0.62	1.11	1.07
MLP DMA Indicator trained on all DMAs	0.86	**0.25**	**0.31**	**1.79**	1.20	**0.71**	**0.78**	0.62	1.10	1.08
MLP trained per DMA	**0.80**	0.26	0.32	1.84	1.31	0.73	0.81	0.63	1.11	1.08
LSTM trained per DMA	0.91	**0.25**	0.34	1.89	**1.17**	0.72	0.92	**0.58**	**1.08**	**1.04**
LSTM trained on all DMAs	0.93	0.27	0.34	1.92	1.46	0.73	0.84	0.65	1.18	1.07
LSTM DMA Indicator trained on all DMAs	1.02	**0.25**	0.35	1.80	1.40	0.74	0.80	0.62	1.12	1.11
Linear Model trained per DMA	0.82	0.31	0.35	1.94	1.79	0.74	0.98	0.81	1.23	1.22
Linear Model trained on all DMAs	0.90	0.31	0.36	1.87	**1.85**	0.72	0.97	0.80	**1.26**	**1.24**
Benchmark Model	**1.07**	**0.36**	**0.40**	**2.11**	1.51	**0.88**	**1.06**	**1.06**	1.24	1.23

Table F.7: GR [-] point forecast of all deterministic models

	A	B	C	D	E	F	G	H	I	J	Average
Linear Model trained on all DMAs	1.19	1.62	**1.08**	1.15	**0.99**	1.09	**1.02**	1.29	**1.11**	**1.12**	**1.17**
Linear Model trained per DMA	**1.24**	**1.59**	**1.08**	1.13	1.04	1.08	1.12	1.35	1.14	**1.12**	1.19
Benchmark Model	1.19	1.90	1.33	**1.10**	1.10	**0.98**	1.05	**1.00**	1.30	1.16	1.21
MLP trained on all DMAs	1.18	1.92	1.21	1.12	1.24	1.09	1.30	1.65	1.29	1.16	1.32
MLP DMA Indicator trained on all DMAs	1.17	2.04	1.25	1.14	1.27	1.09	1.43	1.61	1.32	1.14	1.35
LSTM trained on all DMAs	1.19	2.03	1.29	1.12	1.29	1.16	1.84	1.91	1.52	1.37	1.47
LSTM DMA Indicator trained on all DMAs	**1.04**	2.26	1.38	1.24	1.25	1.15	2.26	2.07	**1.63**	1.24	1.55
MLP trained per DMA	**1.24**	2.28	1.39	1.19	1.73	1.09	2.81	2.47	1.40	1.21	1.68
LSTM trained per DMA	1.20	**2.67**	**1.66**	**1.34**	**1.81**	**1.17**	**3.12**	**2.95**	1.53	**1.55**	**1.90**

F.2. Tables Average Point Forecasts from Probabilistic Models

The models denoted with a * are deterministic trained models.

Table F.8: MAPE [%] point forecasts of all probabilistic models on test set

	A	B	C	D	E	F	G	H	I	J	Average
MCD	14.20	**4.65**	9.46	**6.98**	**1.77**	8.97	3.70	**4.38**	**5.90**	4.65	**6.46**
MLP*	13.38	4.90	9.70	7.04	1.89	**8.80**	4.01	4.50	6.19	**4.64**	6.51
MDN	**13.37**	4.72	9.90	7.16	1.90	8.86	3.78	4.87	6.23	4.71	6.55
Linear model trained per DMA*	13.98	4.87	**9.30**	7.54	2.33	8.96	3.91	4.95	6.08	5.20	6.71
Linear model trained on all DMAs*	15.15	4.81	9.37	7.30	2.25	8.92	**3.53**	4.69	6.13	5.26	6.74
QR	14.07	5.06	9.85	7.25	2.34	8.90	3.96	5.24	6.15	4.77	6.76
CQR	14.30	5.24	10.10	7.41	**2.50**	9.09	**4.22**	**5.35**	6.21	4.95	6.94
Benchmark Model	**18.49**	**6.65**	**12.64**	**8.02**	2.06	**9.71**	4.00	4.80	**7.08**	**5.54**	**7.90**

Table F.9: RMAE [-] point forecasts of probabilistic models on test set

	A	B	C	D	E	F	G	H	I	J	Average
MCD	0.80	**0.70**	**0.72**	**0.87**	**0.87**	0.92	0.91	**0.93**	**0.84**	**0.86**	**0.84**
MLP*	0.80	0.73	0.74	0.88	0.91	**0.90**	1.00	0.95	0.89	**0.86**	0.87
MDN	**0.77**	0.71	0.75	0.89	0.92	0.92	0.96	1.06	**0.91**	0.88	0.88
Linear model trained on all DMAs*	**0.84**	0.72	0.73	0.92	1.10	0.92	**0.89**	0.98	0.87	**0.98**	0.90
Linear model trained per DMA*	0.80	0.72	**0.72**	**0.94**	1.12	0.93	0.98	1.03	0.87	0.96	0.91
QR	0.81	0.75	0.76	0.91	1.13	0.92	0.98	**1.11**	0.89	0.89	0.92
CQR	0.83	**0.78**	**0.78**	0.93	**1.20**	**0.94**	**1.04**	1.09	0.89	0.92	**0.94**

Table F.10: MAE [L/s] point forecasts of all probabilistic models on test set

	A	B	C	D	E	F	G	H	I	J
MCD	1.02	**0.48**	**0.38**	**2.03**	**1.44**	0.79	1.02	**0.98**	**1.37**	**1.23**
MLP*	1.01	0.50	0.39	2.04	1.52	**0.78**	1.11	1.00	1.45	**1.23**
MDN	**0.99**	0.49	0.39	2.08	1.54	0.79	1.07	1.12	1.47	1.26
Linear model trained per DMA*	1.02	0.50	**0.38**	2.20	1.87	0.80	1.09	1.08	1.41	1.37
Linear model trained on all DMAs*	1.07	0.50	0.39	2.14	1.84	0.79	**0.99**	1.03	1.40	1.40
QR	1.03	0.52	0.40	2.12	1.89	0.79	1.10	**1.17**	1.45	1.27
CQR	1.05	0.54	0.41	2.15	**1.99**	0.81	**1.16**	1.15	1.44	1.31
Benchmark Model	**1.27**	**0.69**	**0.53**	**2.33**	1.67	**0.86**	1.11	1.06	**1.62**	**1.43**

Table F.11: MAPE [%] point forecasts of all probabilistic models on validation set

	A	B	C	D	E	F	G	H	I	J	Average
MLP*	11.54	**2.71**	**6.29**	**5.65**	1.54	**9.04**	**3.26**	3.11	5.67	4.21	**5.30**
MDN	**11.15**	2.92	6.33	5.79	1.55	9.06	3.43	3.13	**5.64**	**4.18**	5.32
MCD	11.99	2.86	6.31	5.69	**1.49**	9.17	3.27	**3.08**	5.89	4.24	5.40
QR	11.48	3.11	6.80	5.77	2.20	9.09	3.85	4.14	5.74	4.35	5.65
CQR	11.55	3.26	7.07	5.89	**2.37**	9.24	4.07	4.40	5.88	4.48	5.82
Linear model trained per DMA*	11.17	3.44	7.19	6.00	2.33	9.41	4.03	4.08	6.40	4.74	5.88
Linear model trained on all DMAs*	12.54	3.29	7.12	5.76	2.36	9.24	3.98	4.00	**6.55**	**4.83**	5.97
Benchmark Model	**14.44**	**3.73**	**7.88**	**6.61**	1.96	**11.09**	**4.41**	**5.60**	6.46	4.81	**6.70**

Table F.12: RMAE [-] point forecasts of probabilistic models on validation set

	A	B	C	D	E	F	G	H	I	J	Average
MLP*	0.81	**0.68**	0.79	**0.85**	0.79	**0.81**	**0.73**	**0.59**	**0.88**	**0.88**	**0.78**
MCD	0.81	0.72	**0.78**	0.86	**0.78**	0.82	0.74	**0.58**	0.91	0.88	0.79
MDN	0.78	0.74	0.81	0.88	0.80	**0.81**	0.77	0.59	**0.88**	**0.87**	0.79
QR	0.79	0.78	0.85	0.87	1.11	0.82	0.87	0.77	0.90	0.91	0.87
CQR	0.81	0.81	0.87	0.88	1.19	0.83	0.91	**0.81**	0.92	0.94	0.90
Linear model trained per DMA*	**0.77**	**0.87**	0.89	**0.92**	1.18	**0.84**	**0.92**	0.76	0.99	0.99	0.91
Linear model trained on all DMAs*	**0.84**	0.85	**0.90**	0.88	**1.22**	0.82	**0.91**	0.75	**1.01**	**1.01**	**0.92**

Table F.13: MAE [L/s] point forecasts of all probabilistic models on validation set

	A	B	C	D	E	F	G	H	I	J
MLP*	0.86	**0.25**	**0.31**	**1.79**	1.20	**0.71**	**0.78**	**0.62**	**1.10**	**1.08**
MDN	0.84	0.27	0.32	1.85	1.21	**0.71**	0.82	0.63	**1.10**	**1.08**
MCD	0.87	0.26	**0.31**	1.82	**1.18**	0.72	0.79	**0.62**	1.14	1.09
QR	0.85	0.28	0.34	1.85	1.68	0.72	0.93	0.82	1.12	1.12
CQR	0.87	0.29	0.35	1.87	1.81	0.73	0.97	0.85	1.14	1.16
Linear model trained per DMA*	**0.82**	0.31	0.35	1.94	1.79	0.74	0.98	0.81	1.23	1.22
Linear model trained on all DMAs*	0.90	0.31	0.36	1.87	**1.85**	0.72	0.97	0.80	**1.26**	**1.24**
Benchmark Model	**1.07**	**0.36**	**0.40**	**2.11**	1.51	**0.88**	**1.06**	**1.06**	1.24	1.23

Table F.14: GR [-] point forecasts of all probabilistic models

	A	B	C	D	E	F	G	H	I	J	Average
Linear model trained on all DMAs*	1.19	1.62	**1.08**	**1.15**	**0.99**	1.09	**1.02**	1.29	**1.11**	**1.12**	**1.17**
Linear model trained per DMA*	**1.24**	**1.59**	**1.08**	1.13	1.04	1.08	1.12	1.35	1.14	**1.12**	1.19
Benchmark	1.19	1.90	**1.33**	**1.10**	1.10	**0.98**	1.05	**1.00**	1.30	1.16	1.21
CQR	1.22	1.82	1.18	**1.15**	1.10	1.10	1.20	1.35	1.26	1.14	1.25
QR	1.21	1.85	1.20	**1.15**	1.12	1.10	1.18	1.43	1.30	1.14	1.27
MCD	1.18	1.84	1.22	1.11	1.23	1.10	1.29	1.58	1.20	1.13	1.29
MDN	1.18	1.84	1.22	1.13	**1.27**	**1.11**	1.29	**1.79**	**1.34**	**1.17**	1.33
MLP*	**1.17**	**2.04**	1.25	1.14	**1.27**	1.09	**1.43**	1.61	1.32	1.14	**1.35**

F.3. Figures Point Forecasts over Forecasting Horizon

Figure F.1: Results over forecasting horizon point forecasts of deterministic models (y-axis do not have the same scale)

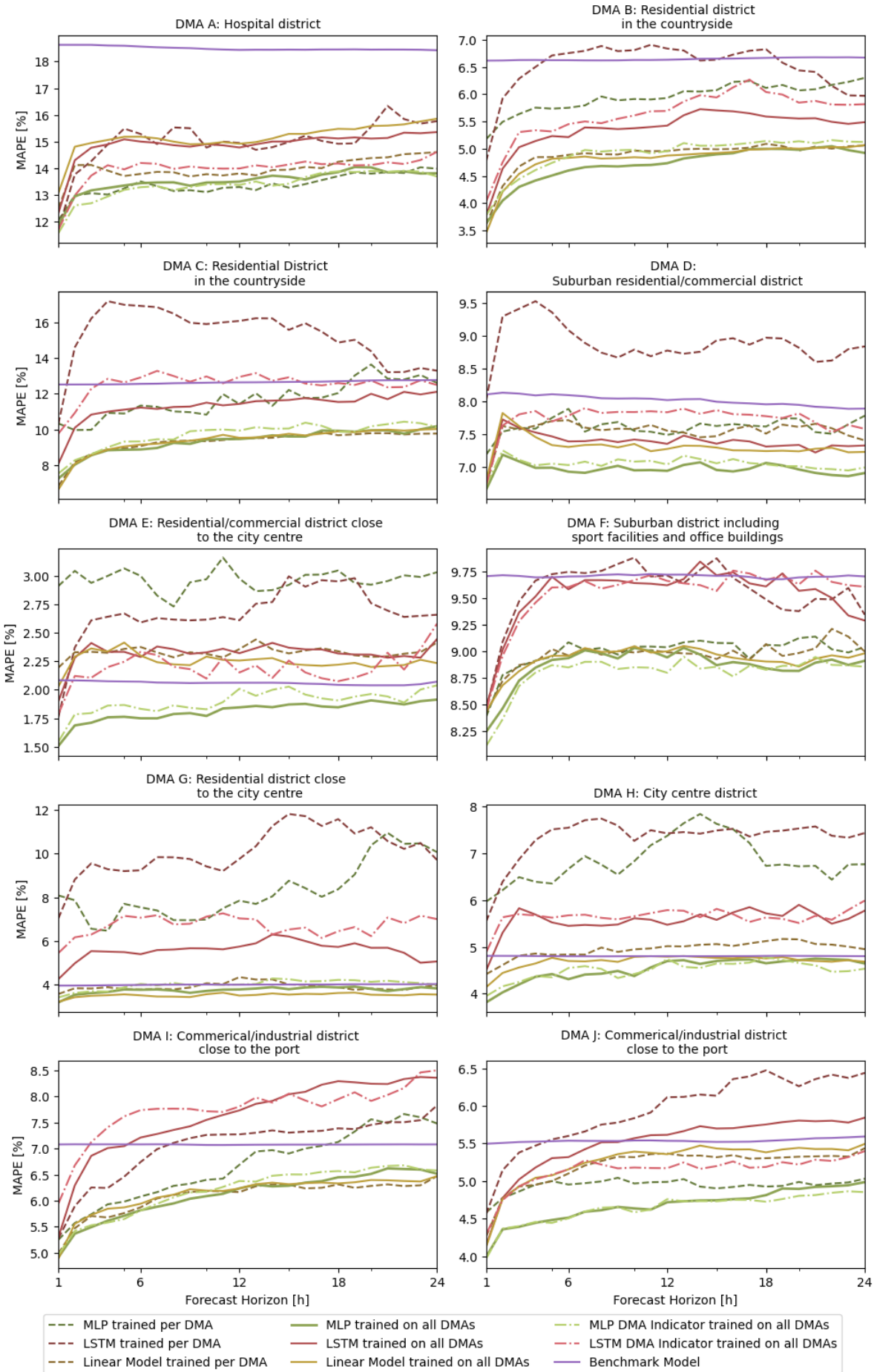


Figure F.2: Results over forecasting horizon point forecasts of probabilistic models (y-axis do not have the same scale) Deterministic models are denoted with a *.

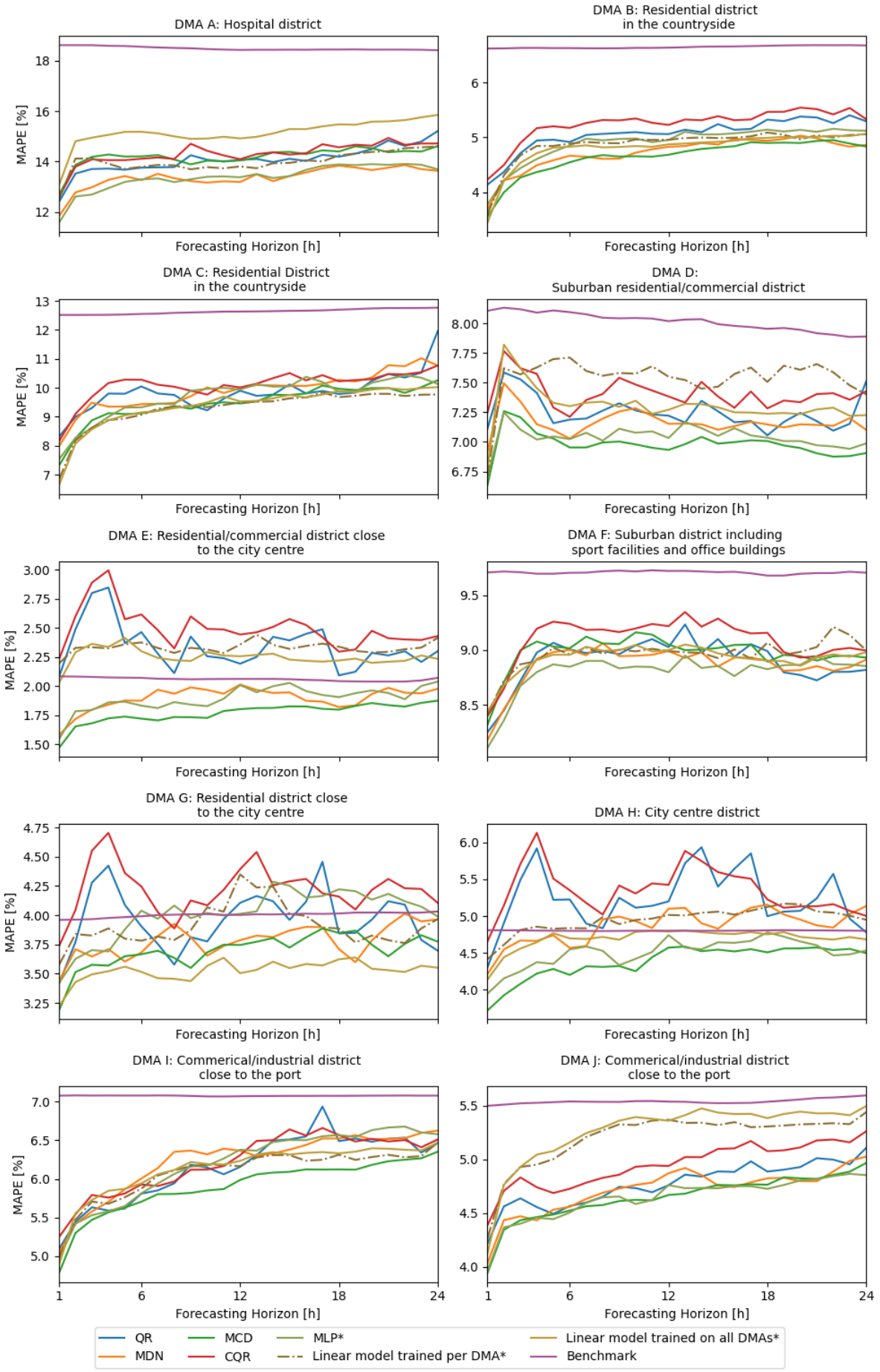


Figure F.3: Average MAPE [-] over all DMAs over the forecasting horizon of deterministic models

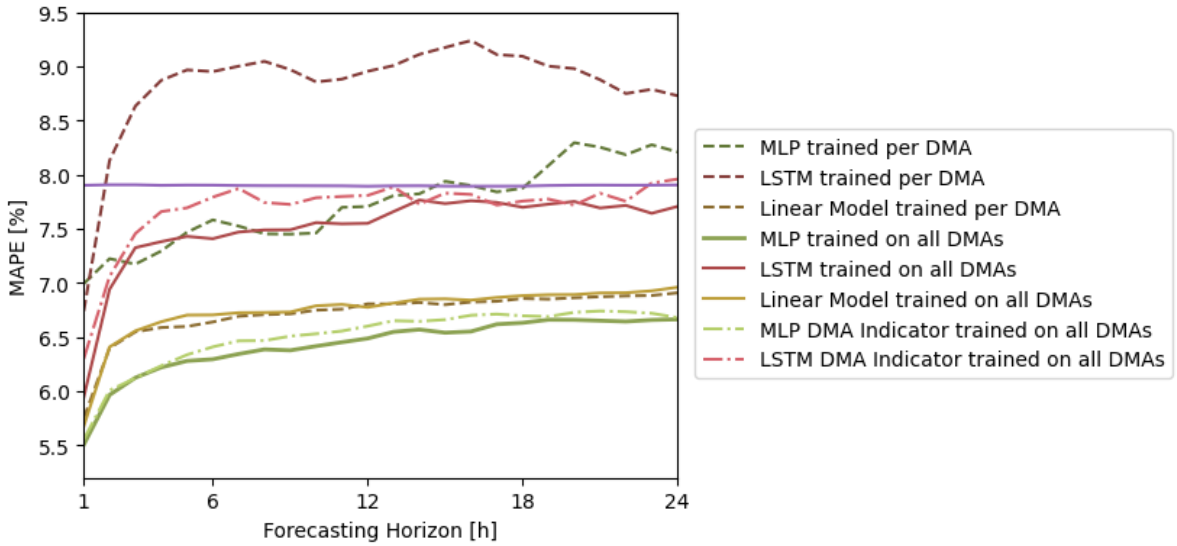
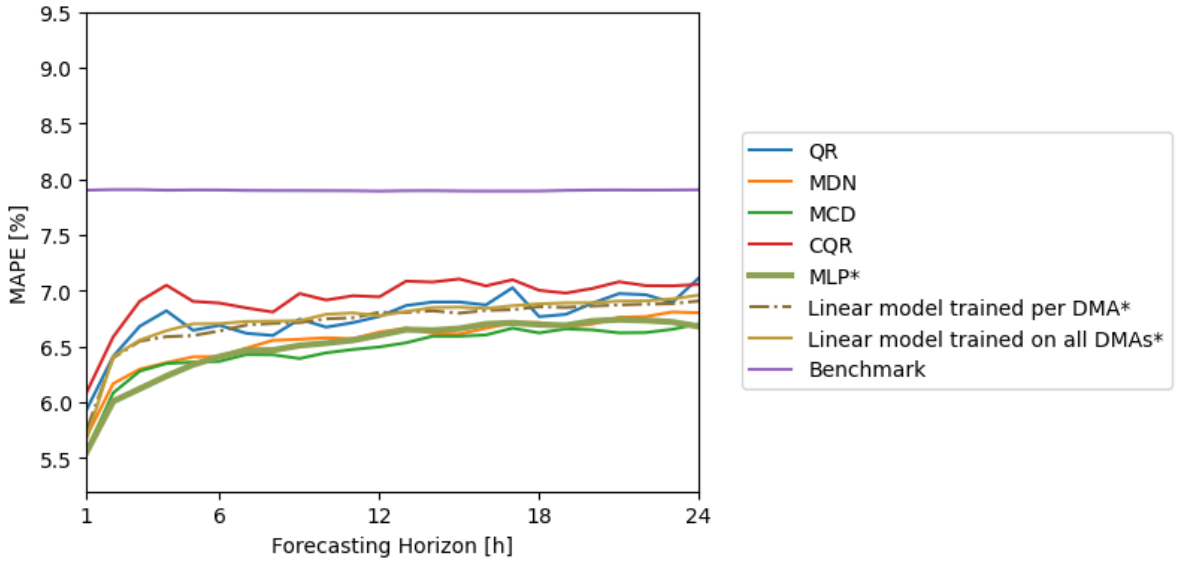


Figure F.4: Average MAPE [-] over all DMAs over the forecasting horizon of probabilistic models. Deterministic models are denoted with a *.



F.4. Tables Probabilistic Forecasts

F.4.1. Coverage on test and validation sets

Table F.15: Coverage of probabilistic models on test set, PICP [-]

	A	B	C	D	E	F	G	H	I	J	Average CG	Max CG
CQR	**0.95**	**0.94**	**0.94**	0.94	**0.95**	**0.95**	0.98	**0.95**	**0.95**	**0.94**	**0.01**	**0.03**
MCD	**0.95**	0.85	0.92	**0.95**	**0.90**	0.96	**0.96**	0.89	**0.95**	**0.94**	0.03	0.10
QR	0.92	**0.94**	0.93	**0.89**	0.99	**0.90**	**0.99**	0.96	0.90	**0.94**	0.03	0.06
MDN	0.94	0.92	0.91	**0.89**	0.98	0.91	**0.99**	0.93	**0.87**	0.92	0.04	0.08
Benchmark Model	**0.91**	**0.83**	**0.90**	0.94	0.94	0.93	**0.96**	**0.88**	**0.87**	**0.87**	**0.05**	**0.12**

Table F.16: Coverage of probabilsitic models on validation set on all DMAs, PICP [-]

	A	B	C	D	E	F	G	H	I	J	Average CG	Max CG
CQR	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.00**	**0.00**
MCD	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.00**	**0.00**
Benchmark Model	**0.94**	**0.95**	0.94	**0.95**	0.97	0.96	0.96	**0.95**	**0.95**	**0.95**	0.01	0.02
MDN	**0.95**	0.96	**0.95**	0.92	0.99	0.93	0.96	0.97	**0.92**	**0.95**	0.02	0.04
QR	**0.94**	**0.99**	**0.97**	**0.91**	**1.00**	**0.91**	**0.98**	**0.99**	**0.92**	**0.95**	**0.03**	**0.05**

F.4.2. Sharpness of Probabilistic models on test and validation set

Table F.17: Sharpness of probabilistic models on test set on all DMAs, PINAW [-]

	A	B	C	D	E	F	G	H	I	J	Average
QR	1.35	1.30	1.17	**0.92**	**0.72**	**1.44**	**1.32**	**0.61**	1.21	0.88	1.09
MDN	**1.32**	1.15	1.05	**0.92**	0.47	1.47	1.18	0.46	**1.12**	**0.84**	**1.00**
MCD	1.42	**0.85**	**1.04**	1.07	**0.29**	**1.89**	**0.79**	**0.36**	**1.58**	0.88	1.02
CQR	1.49	1.31	1.24	1.02	0.44	1.73	1.03	0.50	1.51	0.91	1.12
Benchmark Model	**1.51**	**1.61**	**1.48**	**1.10**	0.50	1.80	1.01	0.42	1.35	**0.93**	**1.17**

Table F.18: Sharpness of probabilistic models on validation set on all DMAs, PINAW [-]

	A	B	C	D	E	F	G	H	I	J	Average
QR	1.33	**1.25**	1.07	**0.83**	**0.60**	**1.32**	0.78	**0.56**	1.10	0.80	0.96
MDN	**1.29**	0.90	**0.89**	0.88	0.36	1.39	0.56	0.37	**1.09**	**0.75**	**0.85**
MCD	1.36	**0.85**	0.93	0.96	**0.25**	1.57	**0.52**	**0.33**	1.27	0.80	0.88
CQR	**1.41**	0.87	0.95	0.95	0.34	1.53	0.60	0.38	1.26	0.81	0.91
Benchmark Model	1.37	1.11	**1.23**	**1.08**	0.48	**1.65**	**0.82**	0.54	**1.38**	**0.89**	**1.06**

F.4.3. Winkler Scores of Probabilistic Forecasts

Table F.19: Winkler Score Test Set on all DMAs [-]

	A	B	C	D	E	F	G	H	I	J
QR	6.55	3.22	2.44	11.69	**16.61**	4.69	**8.74**	**7.34**	9.19	7.53
MDN	**5.98**	**2.95**	**2.16**	11.73	11.06	**4.58**	7.89	6.24	9.46	**7.44**
MCD	**7.62**	3.08	2.30	11.51	**9.38**	5.09	**5.82**	**5.88**	9.30	7.45
CQR	7.51	3.14	2.41	**11.33**	11.39	4.81	6.98	6.34	**8.85**	7.66
Benchmark Model	7.01	**5.28**	**3.34**	**12.46**	14.08	**5.22**	7.22	6.96	**11.08**	**9.81**

Table F.20: Relative Winkler Score Test Set on all DMAs [L/s]

	A	B	C	D	E	F	G	H	I	J	Average
QR	0.93	0.61	0.73	0.94	**1.18**	0.90	**1.21**	**1.05**	0.83	0.77	0.92
MDN	**0.85**	**0.56**	**0.65**	0.94	0.79	**0.88**	1.09	0.90	0.85	**0.76**	0.83
MCD	**1.09**	0.58	0.69	0.92	**0.67**	0.98	**0.81**	**0.84**	0.84	**0.76**	**0.82**
CQR	1.07	0.60	0.72	**0.91**	0.81	0.92	0.97	0.91	**0.80**	0.78	0.85
Benchmark Model	1.00	**1.00**	**1.00**	**1.00**	1.00	**1.00**	1.00	1.00	**1.00**	**1.00**	**1.00**

Table F.21: Conditional Relative Winkler Score on test set by accepting a coverage drop to 0.94

	A	B	C	D	E	F	G	H	I	J
QR	-	**0.61**	-	-	**1.18**	-	**1.21**	**1.05**	-	-
MDN	-	-	-	-	**0.79**	-	1.09	-	-	-
MCD	**1.09**	-	-	**0.92**	-	**0.98**	**0.81**	-	**0.84**	**0.76**
CQR	**1.07**	**0.60**	**0.72**	-	0.81	**0.92**	0.97	**0.91**	**0.80**	**0.78**
Benchmark Model	-	-	-	-	-	-	1.00	-	-	-

Table F.22: Conditional Relative Winkler Score on test set by accepting a coverage drop to 0.93

	A	B	C	D	E	F	G	H	I	J
QR	-	**0.61**	-	-	**1.18**	-	**1.21**	**1.05**	-	0.77
MDN	**0.85**	-	-	-	**0.79**	-	1.09	-	-	-
MCD	**1.09**	-	-	0.92	-	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	**0.60**	**0.72**	**0.91**	0.81	**0.92**	0.97	**0.91**	**0.80**	**0.78**
Benchmark Model	-	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

Table F.23: Conditional Relative Winkler Score on test set by accepting a coverage drop to 0.92

	A	B	C	D	E	F	G	H	I	J
QR	0.93	**0.61**	**0.73**	-	**1.18**	-	**1.21**	**1.05**	-	0.77
MDN	**0.85**	**0.56**	-	-	**0.79**	-	1.09	**0.90**	-	**0.76**
MCD	**1.09**	-	-	0.92	-	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	0.60	**0.72**	**0.91**	0.81	**0.92**	0.97	0.91	**0.80**	**0.78**
Benchmark Model	-	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

Table F.24: Conditional Relative Winkler Score on test set by accepting a coverage drop to 0.91

	A	B	C	D	E	F	G	H	I	J
QR	0.93	**0.61**	**0.73**	-	**1.18**	-	**1.21**	**1.05**	-	0.77
MDN	**0.85**	**0.56**	**0.65**	-	**0.79**	**0.88**	1.09	**0.90**	-	**0.76**
MCD	**1.09**	-	0.69	0.92	-	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	0.60	0.72	**0.91**	0.81	0.92	0.97	0.91	**0.80**	**0.78**
Benchmark Model	1.00	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

Table F.25: Conditional Relative Winkler Score on test set by accepting a coverage drop to 0.90

	A	B	C	D	E	F	G	H	I	J
QR	0.93	**0.61**	**0.73**	-	**1.18**	0.90	**1.21**	**1.05**	-	0.77
MDN	**0.85**	**0.56**	**0.65**	-	0.79	**0.88**	1.09	**0.90**	-	**0.76**
MCD	**1.09**	-	0.69	0.92	**0.67**	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	0.60	0.72	**0.91**	0.81	0.92	0.97	0.91	**0.80**	**0.78**
Benchmark Model	1.00	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

Table F.26: Conditional Relative Winkler Score on test set by accepting a coverage drop of 0.01 relative to the coverage of the validation set

	A	B	C	D	E	F	G	H	I	J
QR	-	**0.61**	-	-	**1.18**	-	**1.21**	**1.05**	-	-
MDN	-	-	-	-	**0.79**	-	1.09	-	-	-
MCD	**1.09**	-	-	**0.92**	-	**0.98**	**0.81**	-	**0.84**	**0.76**
CQR	**1.07**	**0.60**	**0.72**	-	0.81	**0.92**	0.97	**0.91**	**0.80**	**0.78**
Benchmark Model	-	-	-	-	-	-	1.00	-	-	-

Table F.27: Conditional Relative Winkler Score on test set by accepting a coverage drop of 0.02 relative to the coverage of the validation set

	A	B	C	D	E	F	G	H	I	J
QR	-	**0.61**	-	-	**1.18**	-	**1.21**	**1.05**	-	0.77
MDN	**0.85**	-	-	-	**0.79**	-	1.09	-	-	-
MCD	**1.09**	-	-	0.92	-	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	**0.60**	**0.72**	**0.91**	0.81	**0.92**	0.97	**0.91**	**0.80**	**0.78**
Benchmark Model	-	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

Table F.28: Conditional Relative Winkler Score on test set by accepting a coverage drop of 0.03 relative to the coverage of the validation set

	A	B	C	D	E	F	G	H	I	J
QR	0.93	**0.61**	**0.73**	-	**1.18**	-	**1.21**	**1.05**	-	0.77
MDN	**0.85**	**0.56**	-	-	**0.79**	-	1.09	**0.90**	-	**0.76**
MCD	**1.09**	-	-	0.92	-	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	0.60	**0.72**	**0.91**	0.81	**0.92**	0.97	0.91	**0.80**	**0.78**
Benchmark Model	-	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

Table F.29: Conditional Relative Winkler Score on test set by accepting a coverage drop of 0.04 relative to the coverage of the validation set

	A	B	C	D	E	F	G	H	I	J
QR	0.93	**0.61**	**0.73**	-	**1.18**	-	**1.21**	**1.05**	-	0.77
MDN	**0.85**	**0.56**	**0.65**	-	**0.79**	**0.88**	1.09	**0.90**	-	**0.76**
MCD	**1.09**	-	0.69	0.92	-	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	0.60	0.72	**0.91**	0.81	0.92	0.97	0.91	**0.80**	**0.78**
Benchmark Model	1.00	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

Table F.30: Conditional Relative Winkler Score on test set by accepting a coverage drop of 0.05 relative to the coverage of the validation set

	A	B	C	D	E	F	G	H	I	J
QR	0.93	**0.61**	**0.73**	-	**1.18**	0.90	**1.21**	**1.05**	-	0.77
MDN	**0.85**	**0.56**	**0.65**	-	0.79	**0.88**	1.09	**0.90**	-	**0.76**
MCD	**1.09**	-	0.69	0.92	**0.67**	0.98	**0.81**	-	**0.84**	**0.76**
CQR	1.07	0.60	0.72	**0.91**	0.81	0.92	0.97	0.91	**0.80**	**0.78**
Benchmark Model	1.00	-	-	**1.00**	1.00	**1.00**	1.00	-	-	-

F.5. Figures Probabilistic Forecasts

Figure F.5: Probabilistic models coverage versus sharpness over horizon on all DMAs and testing data, first step of horizon is a dot, mean predictions have a cross

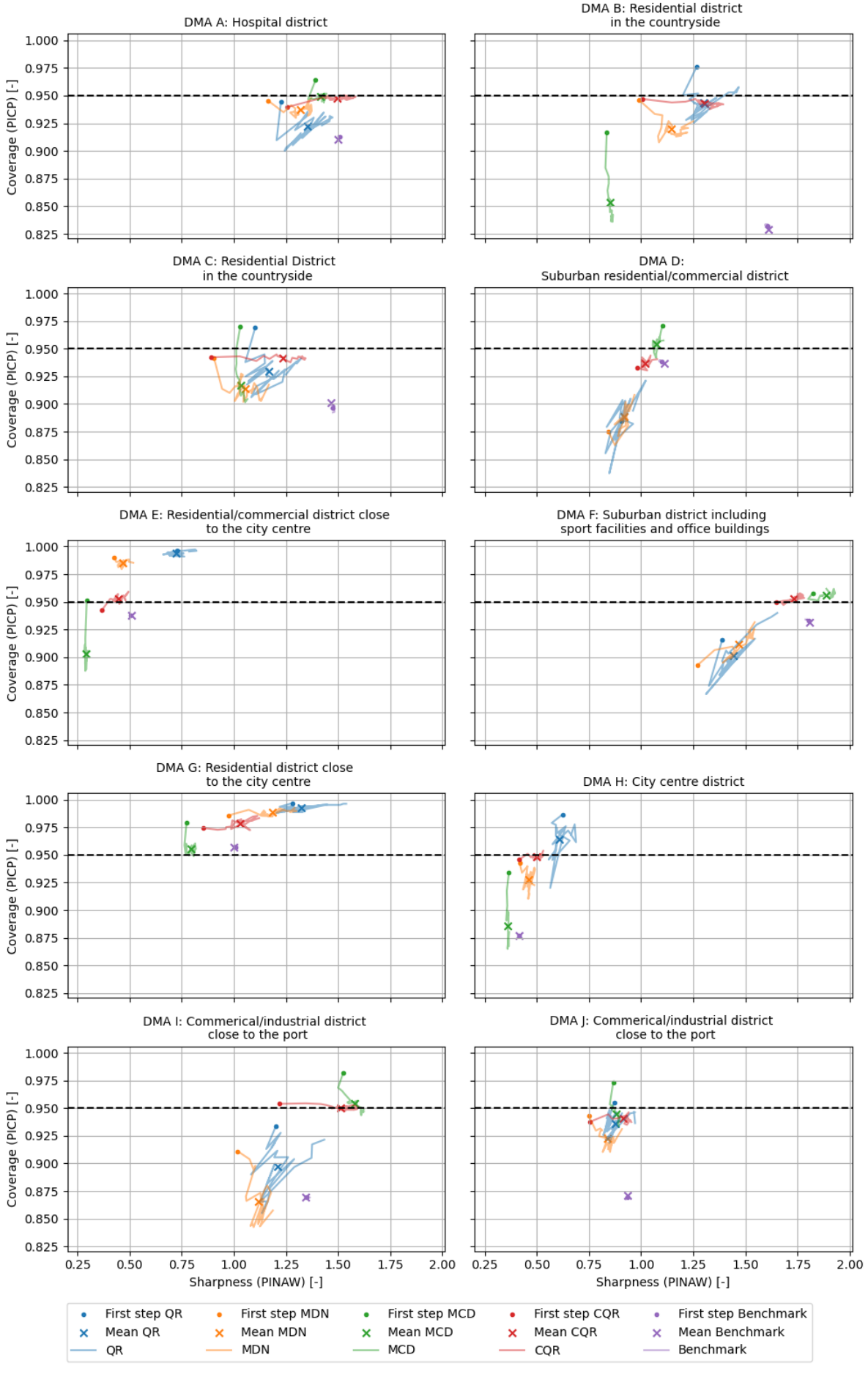


Figure F.6: Probabilistic models coverage versus sharpness over horizon on all DMAs and validation data, first step of horizon is a dot, mean predictions have a cross

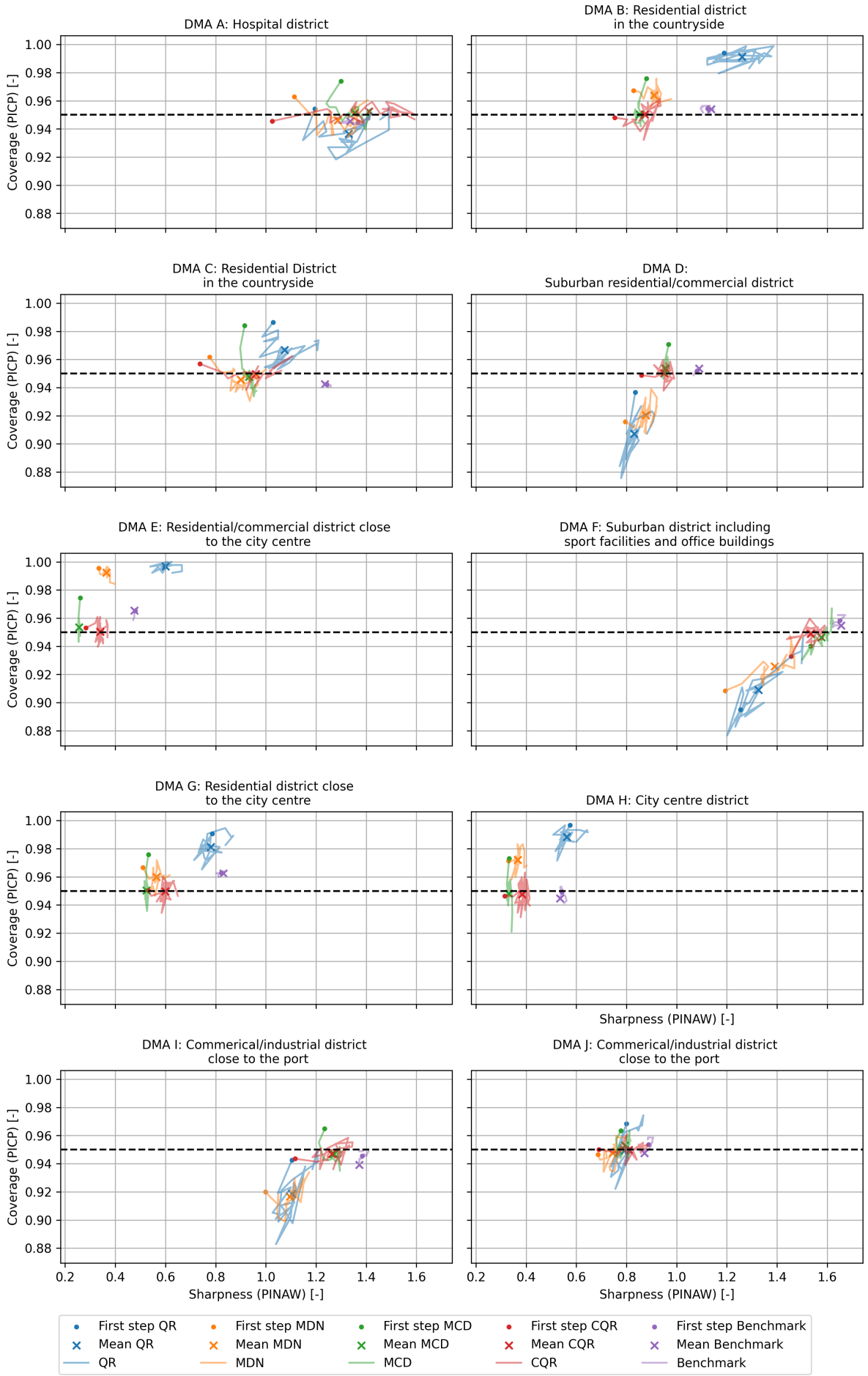
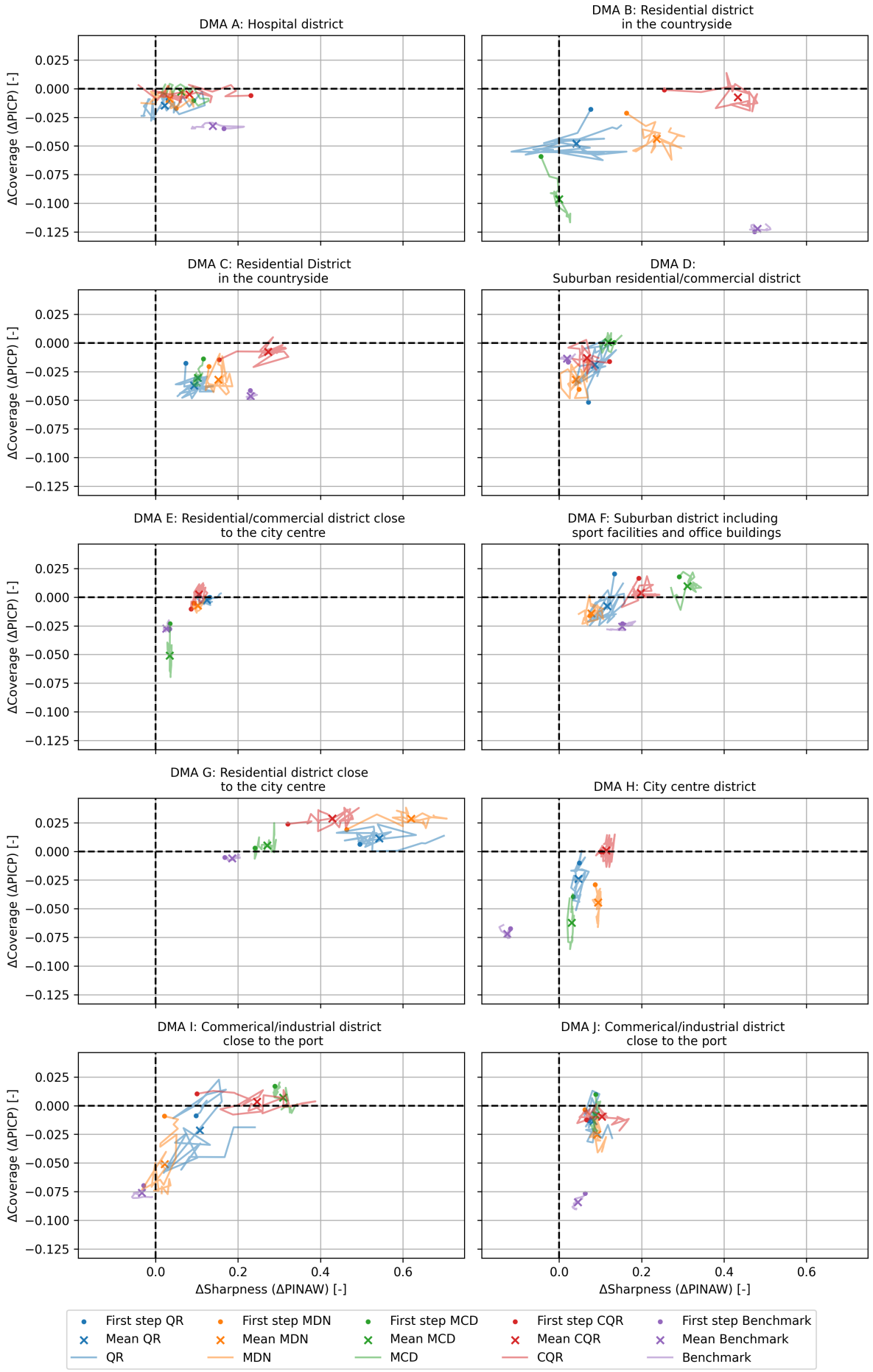


Figure F.7: Probabilistic models delta coverage versus delta sharpness over horizon on all DMAs, delta is between testing and validation data, first step of horizon is a dot, mean predictions have a cross



F.5.1. Rolling Coverage

Figure F.8: Rolling coverage of all models on all DMAs

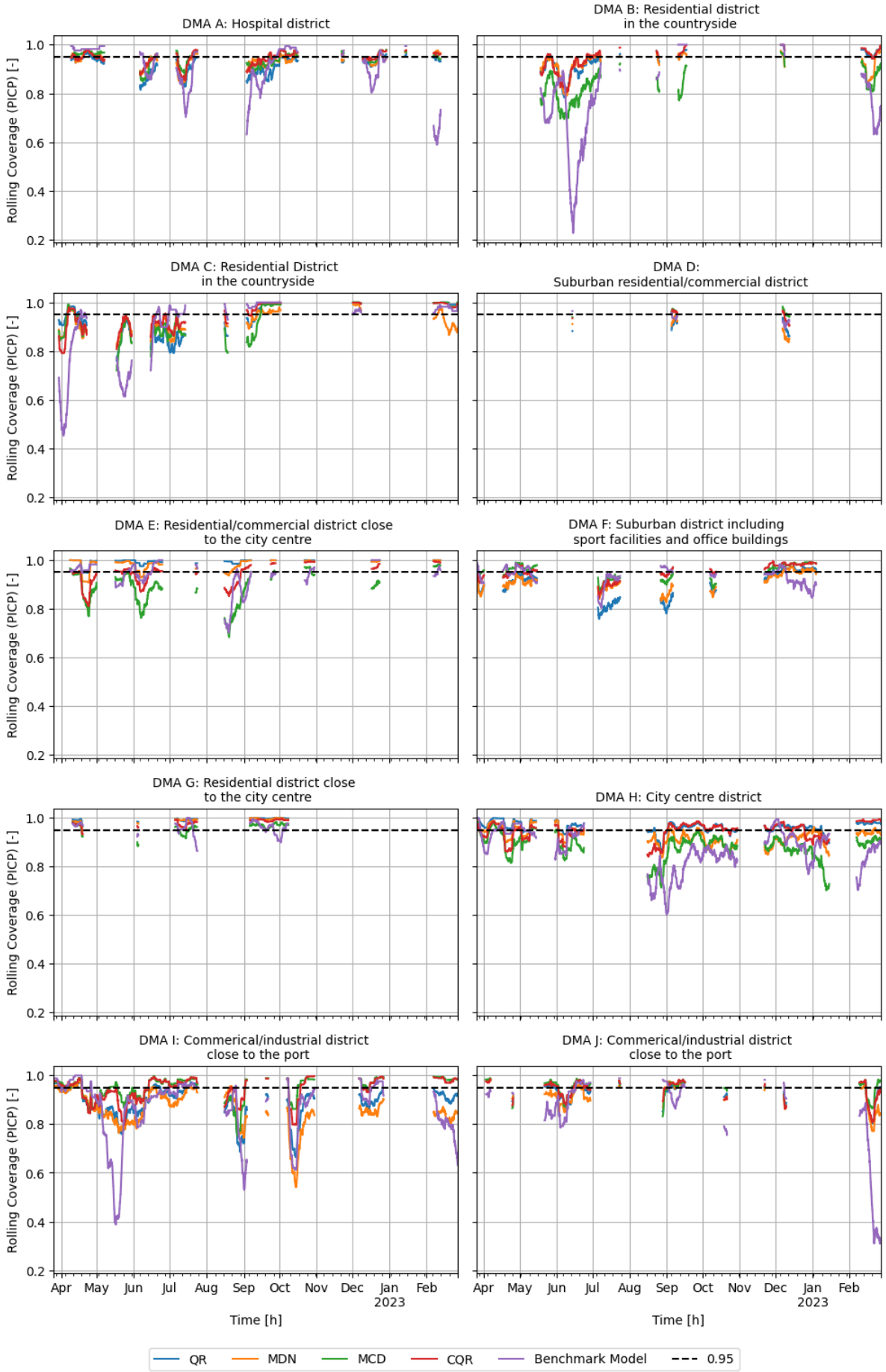


Figure F.9: Rolling coverage of all models on all DMAs besides Benchmark Model

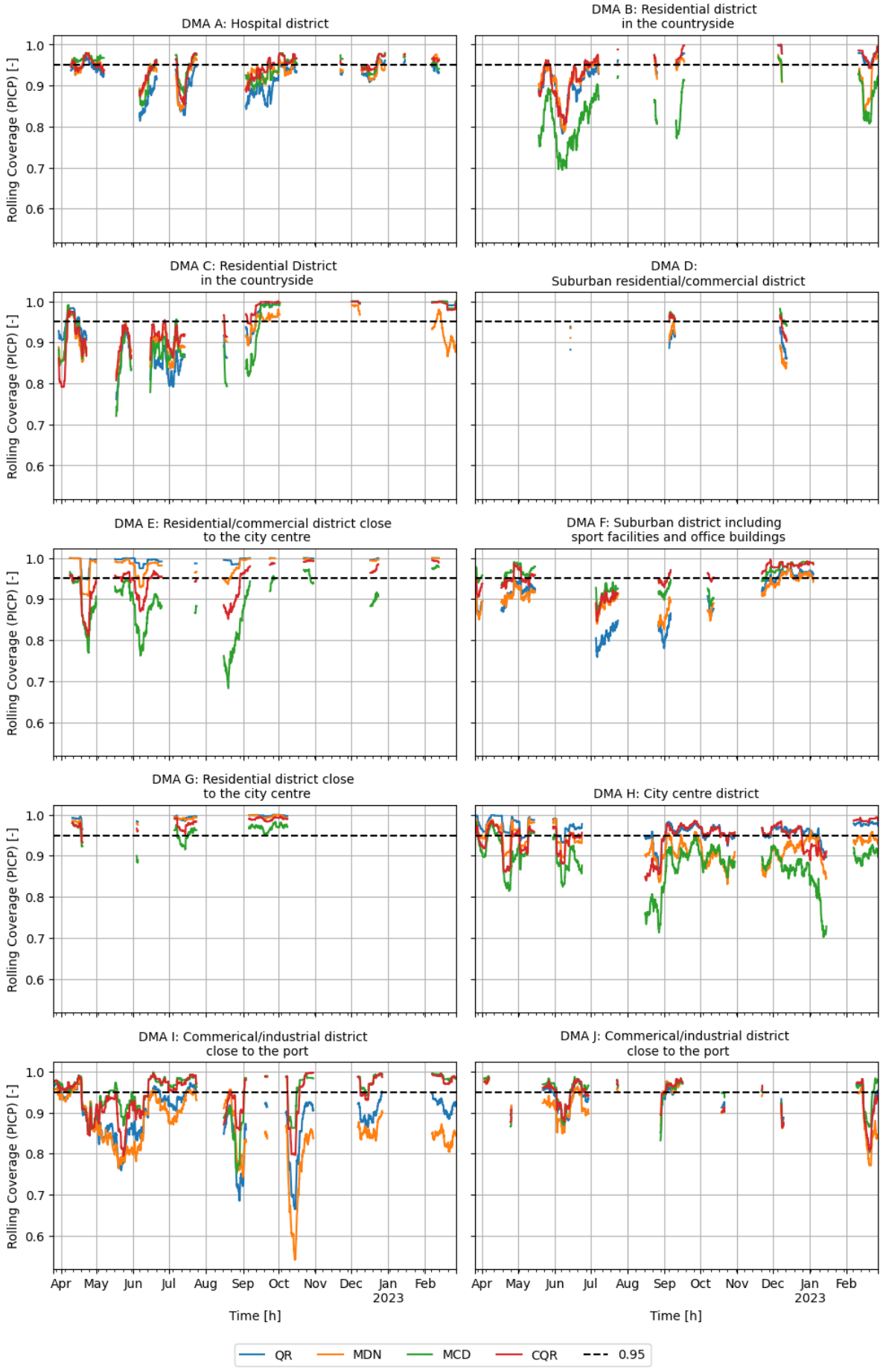


Table F.31: Median and Mean Statistics of Rolling PICP on all DMAs [-]

Statistic					Median					Mean
DMA	QR	MDN	MCD	CQR	Benchmark	QR	MDN	MCD	CQR	Benchmark
A	0.93	0.94	**0.95**	0.95	0.94	0.92	0.93	0.94	**0.94**	0.91
B	0.94	0.93	0.84	**0.95**	0.78	0.93	0.92	0.84	**0.93**	0.75
C	**0.95**	0.91	0.93	0.96	0.96	0.93	0.92	0.93	**0.95**	0.90
D	0.91	0.88	0.96	**0.96**	0.93	0.90	0.89	0.96	**0.94**	0.92
E	1.00	0.99	0.90	0.95	**0.95**	0.99	0.98	0.89	**0.94**	0.94
F	0.92	0.91	0.97	**0.96**	0.94	0.90	0.91	0.96	**0.95**	0.93
G	1.00	0.99	0.97	0.99	**0.96**	0.99	0.99	0.96	0.98	**0.96**
H	0.97	0.93	0.89	**0.95**	0.89	0.96	0.93	0.88	**0.95**	0.88
I	0.91	0.86	0.97	**0.96**	0.90	0.89	0.86	**0.95**	0.94	0.86
J	**0.94**	0.92	0.96	0.96	0.89	0.93	0.92	**0.95**	0.94	0.84
Average CG	0.03	0.04	0.03	**0.01**	0.04	0.03	0.04	0.03	**0.01**	0.06
Median CG	0.03	0.04	0.02	**0.01**	0.02	0.04	0.03	0.01	**0.01**	0.05
Min	0.91	0.86	0.84	**0.95**	0.78	0.89	0.86	0.84	**0.93**	0.75
Max	1.00	0.99	0.97	0.99	**0.96**	0.99	0.99	0.96	0.98	**0.96**

Table F.32: Minimum and Maximum Statistics of Rolling PICP on all DMAs [-]

Statistic					Min					Max
DMA	QR	MDN	MCD	CQR	Benchmark	QR	MDN	MCD	CQR	Benchmark
A	0.81	0.84	0.85	**0.85**	0.59	**0.97**	0.98	0.98	0.98	0.99
B	0.78	0.79	0.69	**0.81**	0.23	1.00	0.98	**0.97**	1.00	1.00
C	0.76	**0.82**	0.72	0.79	0.45	1.00	**0.99**	1.00	1.00	1.00
D	0.86	0.84	**0.94**	0.90	0.89	0.94	**0.95**	0.98	0.97	0.96
E	**0.95**	0.91	0.68	0.81	0.70	1.00	1.00	**0.99**	0.99	1.00
F	0.76	0.83	**0.87**	0.85	0.80	**0.97**	0.97	0.99	1.00	1.00
G	0.96	**0.95**	0.88	0.93	0.86	1.00	1.00	**0.98**	1.00	1.00
H	**0.87**	0.83	0.70	0.84	0.60	1.00	0.99	**0.98**	0.99	1.00
I	0.66	0.54	0.75	**0.79**	0.39	**0.98**	0.98	1.00	1.00	1.00
J	0.80	0.77	**0.83**	0.81	0.31	0.99	0.99	0.99	**0.98**	0.99
Average CG	0.13	0.14	0.16	**0.11**	0.37	0.04	**0.03**	0.04	0.04	0.04
Median CG	0.15	**0.12**	0.20	0.14	0.36	0.05	0.04	**0.04**	0.05	0.05
Min	0.66	0.54	0.68	**0.79**	0.23	0.94	**0.95**	0.97	0.97	0.96
Max	0.96	**0.95**	0.94	0.93	0.89	**1.00**	1.00	1.00	1.00	1.00