

Article

Probabilistic Deep Learning to Quantify Uncertainty in Air Quality Forecasting

Abdulmajid Murad ^{1,*} , Frank Alexander Kraemer ¹ , Kerstin Bach ²  and Gavin Taylor ³ 

¹ Department of Information Security and Communication Technology, Norwegian University of Science and Technology, 7491 Trondheim, Norway; kraemer@ntnu.no

² Department of Computer Science, Norwegian University of Science and Technology, 7491 Trondheim, Norway; kerstin.bach@ntnu.no

³ Department of Computer Science, United States Naval Academy, Annapolis, MD 21402, USA; taylor@usna.edu

* Correspondence: abdulmajid.a.murad@ntnu.no

Abstract: Data-driven forecasts of air quality have recently achieved more accurate short-term predictions. However, despite their success, most of the current data-driven solutions lack proper quantifications of model uncertainty that communicate how much to trust the forecasts. Recently, several practical tools to estimate uncertainty have been developed in probabilistic deep learning. However, there have not been empirical applications and extensive comparisons of these tools in the domain of air quality forecasts. Therefore, this work applies state-of-the-art techniques of uncertainty quantification in a real-world setting of air quality forecasts. Through extensive experiments, we describe training probabilistic models and evaluate their predictive uncertainties based on empirical performance, reliability of confidence estimate, and practical applicability. We also propose improving these models using “free” adversarial training and exploiting temporal and spatial correlation inherent in air quality data. Our experiments demonstrate that the proposed models perform better than previous works in quantifying uncertainty in data-driven air quality forecasts. Overall, Bayesian neural networks provide a more reliable uncertainty estimate but can be challenging to implement and scale. Other scalable methods, such as deep ensemble, Monte Carlo (MC) dropout, and stochastic weight averaging-Gaussian (SWAG), can perform well if applied correctly but with different tradeoffs and slight variations in performance metrics. Finally, our results show the practical impact of uncertainty estimation and demonstrate that, indeed, probabilistic models are more suitable for making informed decisions.



Citation: Murad, A.; Kraemer, F.A.; Bach, K.; Taylor, G. Probabilistic Deep Learning to Quantify Uncertainty in Air Quality Forecasting. *Sensors* **2021**, *21*, 8009. <https://doi.org/10.3390/s21238009>

Academic Editor: Yeh Hsi-Jen James

Received: 4 November 2021

Accepted: 25 November 2021

Published: 30 November 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: air quality; probabilistic deep learning; epistemic uncertainty; aleatoric uncertainty

1. Introduction

Monitoring and forecasting real-world phenomena are fundamental use cases for many practical applications in the Internet of Things (IoT). For example, policymakers in municipalities can use forecasts of ambient air quality to make decisions about actions, such as informing the public or starting emission-reduction measures. The problem is that forecasts are both uncertain and intended for human interpretation, so decisions about specific actions should take forecast confidence into account. For instance, it may be best to only start a costly initiative for cleaning streets of dust when the forecast of air pollutants exceeds a certain threshold and the reported confidence is high. Therefore, quantifying the predictive confidence is crucial for learning, providing, and interpreting reliable forecasting models.

Progress in probabilistic machine learning [1] and, more recently, in probabilistic deep learning led to the development of practical tools to estimate uncertainty about models and predictions [2–7]. These tools have been successfully used in various domains, such as computer vision [8,9], language modeling [10,11], machine translation [12]

and autonomous driving [13]. All of these tools address quantifying predictive uncertainty but differ in techniques, approximations, and assumptions when representing and manipulating uncertainty.

The successful application of probabilistic models to real human problems requires us to bridge the gap from the theory of these disparate approaches to practical concerns. In particular, we need an empirical evaluation and comparison of these many techniques. We want to know how they perform with respect to prediction accuracy, uncertainty quantification, and other requirements. Specifically, we want to know the reliability of their confidence estimate, meaning if they are actually more accurate and trustworthy when their confidence is high. This is practically important for policymakers to make risk-informed decisions. Such an empirical evaluation, especially in the domain of air quality, is currently lacking.

Existing studies already incorporate aspects of uncertainty into their data-driven forecasts, but they often only quantify aleatoric uncertainty. This uncertainty is inherent in the observed data and can be quantified by a distribution over the model's output using softmax or Gaussian. There is also epistemic uncertainty that we can and should quantify as well. This type of uncertainty represents how much the model does not know, for instance, in regions of the input space with little data. Probabilistic models can capture both types of uncertainties, which makes them practically appealing since they convey more information about the reliability of the forecast. They also provide a wider area of control over the decision boundary and the risk profile of a decision. For instance, Figure 1 illustrates the potential of quantifying both types of uncertainties in decision making. It shows the decision F1 score as a function of normalized aleatoric and epistemic confidence thresholds in a probabilistic model. Each point on the surface represents the resulting score of accepting the model predictions only where its confidence is above specific thresholds (τ_1 and τ_2). Expressing both uncertainty dimensions gives more control regarding false positives and false negatives and their associated risks. We will come back to this Figure in Section 5.3.

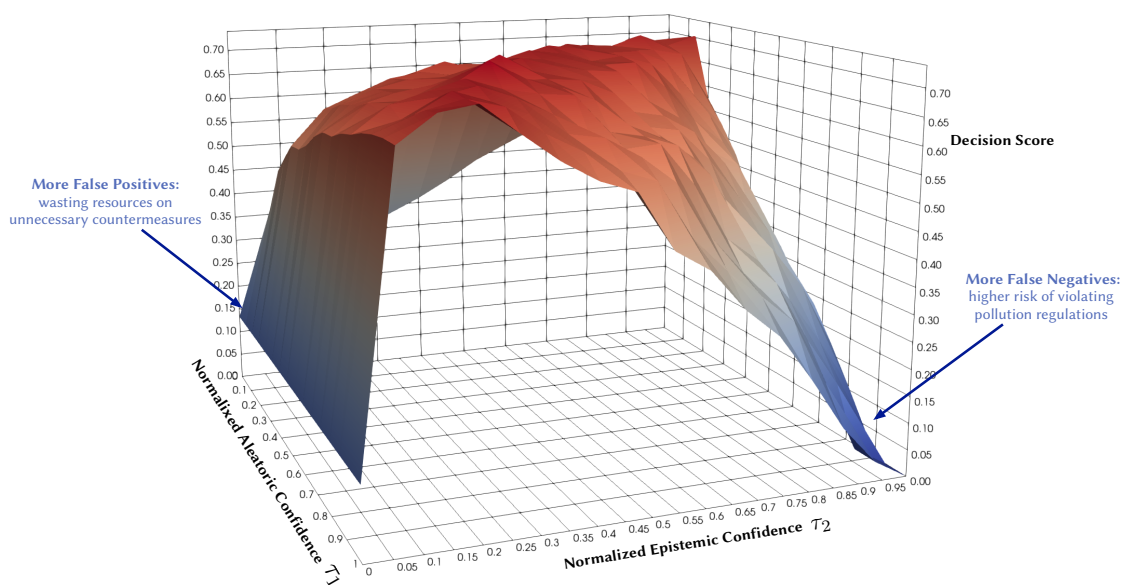


Figure 1. Illustrative example: decision F1 score as a function of normalized aleatoric and epistemic confidence thresholds.

In this work, we develop a set of deep probabilistic models for air quality forecasting that quantify both aleatoric and epistemic uncertainties and study how to represent and manipulate their predictive uncertainties. Our contributions are the following:

- We conduct a broad empirical comparison and exploratory assessment of state-of-the-art techniques in deep probabilistic learning applied to air quality forecasting. Through exhaustive experiments, we describe training these models and evaluat-

ing their predictive uncertainties using various metrics for regression and classification tasks.

- We improve uncertainty estimation using adversarial training to smooth the conditional output distribution locally around training data points.
- We apply uncertainty-aware models that exploit the temporal and spatial correlation inherent in air quality data using recurrent and graph neural networks.
- We introduce a new state-of-the-art example for air quality forecasting by defining the problem setup and selecting proper input features and models.

Our results show that all the considered probabilistic models perform well on our application, with slight variations in different performance metrics. Bayesian neural networks perform slightly better in proper scoring rules that measure the quality of probabilistic predictions. In addition, we show that smoothing predictive distributions by adversarial training improves metrics that punish incorrect, overconfident predictions, especially in regression tasks, since the forecasted phenomena are inherently smooth.

The rest of this paper is organized as follows: In Section 2, we outline the related work in uncertainty estimation and air quality forecasting. We then define the problem setup and introduce non-probabilistic baselines and evaluation metrics in Section 3. In Section 4, we present deep probabilistic models applied to the air quality forecast, describe their training, evaluation, and how to improve their uncertainty estimate. We close with a comparative analysis to compare the selected models in Section 5 and a conclusion in Section 6. Code and dataset are available at https://github.com/Abdulmajid-Murad/deep_probabilistic_forecast (accessed on 27 November 2021).

2. Related Work

Forecasting ambient air quality is crucial to support decision-making in urban management. Therefore, a sizeable body of work addresses building air quality forecasting models. For example, MACC (Monitoring Atmospheric Composition and Climate) [14] is a European project that provides air quality analysis and forecasting services for the European Continent. It uses physics-based modeling that combines transport, chemistry, and satellite data to provide a multi-model ensemble forecast of atmospheric composition (daily forecasts with hourly outputs of 10 chemical species/aerosols). Walker et al. [15] used the output of MACC ensemble-based probabilistic forecasting of air pollution and performed statistical post-processing, calibrated predictive distribution using Box-Cox transformation for correcting the skewness of air pollution data. Additionally, they discussed model selection and verification using Akaike and Bayesian information criteria. To obtain the ensemble forecast from MACC, they introduced stochastic perturbations to the emissions. Garaud et al. [16] performed a posterior calibration of multi-model ensembles using a mixed optimization algorithm to extract a sub-ensemble.

The official air quality forecasting service in Norway [17] uses a Gaussian dispersion modeling that provides a 2-day hourly forecast with high-resolution coverage (between 250 and 50 m grid) over the entire country [18,19]. The forecast is based on weather conditions, polluting emissions, and terrain. In particular, it uses the chemical transport model uEMEP (urban European Monitoring and Evaluation Program) [18] and a road dust emission model [20]. Denby et al. [21] analyze the accuracy of the Norwegian air quality forecasting by comparing model calculations with measurements. They show that the model's forecast on particle dust is marginally better than the persistent forecast and give some assumptions about why model calculations deviate from the observations.

Although physics-based models [18,19,22] can provide long-range air pollution information, they require significant domain knowledge and complex modeling of dispersion, chemical transport, and meteorological processes. Additionally, physics-based models involve structural uncertainty, low spatial resolution, and do not capture abrupt and short-term changes in air pollution. Data-driven modeling based on historical data [23–25] can complement physics-based modeling by learning directly from air quality measurements and providing a more reliable short-term prediction. For example, Lepperod et al. [23]

deployed stationary and mobile micro-sensor devices and used Narrowband IoT (NB-IoT) to aggregate air quality data from these sensors. Then, they applied machine learning methods to predict air quality in the next 48 h using observations of sensors' measurements, traffic, and weather data. Zhou et al. [25] used long short-term memory (LSTM) to forecast multi-step time-series of air quality, while Mokhtari et al. [26] proposed combining a convolutional neural network (CNN) with LSTM for air quality prediction and quantified the uncertainty using quantile regression and MC dropout. Tao et al. [27] used 1D CNNs and a Bidirectional gated recurrent unit (GRU) for a short-term forecast of fine air particles. Pucer et al. [28] used Gaussian Processes (GP) to forecast daily air-pollutant levels, and Aznarte et al. [29] proposed using quantile regression for probabilistic forecasting of extreme nitrogen dioxide (NO_2) pollution.

Most of the current data-driven forecasts give point predictions of a deterministic nature. Thus, they lack useful estimates of their predictive uncertainty that convey more information about how much to trust the forecast. Recently, quantifying prediction uncertainty has garnered increasing attention in machine learning fields, including deep learning. Bayesian methods are among the most used approaches for uncertainty estimation in neural networks. Given a prior distribution over the parameters, Bayesian methods use training data to compute a posterior distribution. Using the obtained distribution, we can easily quantify the predictive uncertainty. This approach has been extended to neural networks, theoretically allowing for the accuracy of modern prediction methods with valid uncertainty measures; however, modern neural networks contain many parameters, and obtaining explicit posterior densities through Bayesian inference is computationally intractable. Instead, there exist a variety of approximation methods that estimate the posterior distributions. These methods can be decomposed into three main categories, either based on variational inference [3,30,31], Markov chain Monte Carlo (MCMC) [32–34], or Laplace approximation [2,35]. In this paper, we will use variational Bayesian methods and approximate Bayesian inference methods.

3. Air Quality Prediction, Base Models and Metrics

3.1. Problem Setup

We are trying to build a model for forecasting air quality trends at pre-defined locations and for a specified forecast horizon. In our case study, we want to forecast the level of microscopic particles in the air, known as particulate matter (PM). These are inhalable particles with two types: coarse particles with a diameter less than $10\ \mu\text{m}$ (PM_{10}) and fine particles with a diameter less than $2.5\ \mu\text{m}$ ($PM_{2.5}$). The forecast predicts pollutant levels for the next 24 h at four monitoring stations in the city of Trondheim, as shown in Figure 2. The stakeholders, policymakers of the municipality, would like to estimate if the concentration of air particles exceeds certain thresholds following the Common Air Quality Index (CAQI) used in Europe [36], as shown in Table 1. This can be achieved through value regression or by directly classifying threshold exceedance levels. We will explore probabilistic models that forecast air quality values and predict threshold exceedance events. Using probabilistic models provides more qualitative information since decision-making largely depends on the credibility intervals of specific predictions. The forecast is based on explanatory variables, such as historical air quality measurements, meteorological data, traffic, and street-cleaning reports from the municipality.

Table 1. European Common Air Quality Index.

Index	$PM_{10}(\mu\text{g}/\text{m}^3)$	$PM_{10}(\mu\text{g}/\text{m}^3)$
Very low	0–25	0–15
Low	25–50	15–30
Medium	50–90	30–55
High	90–180	55–110
Very High	>180	>110

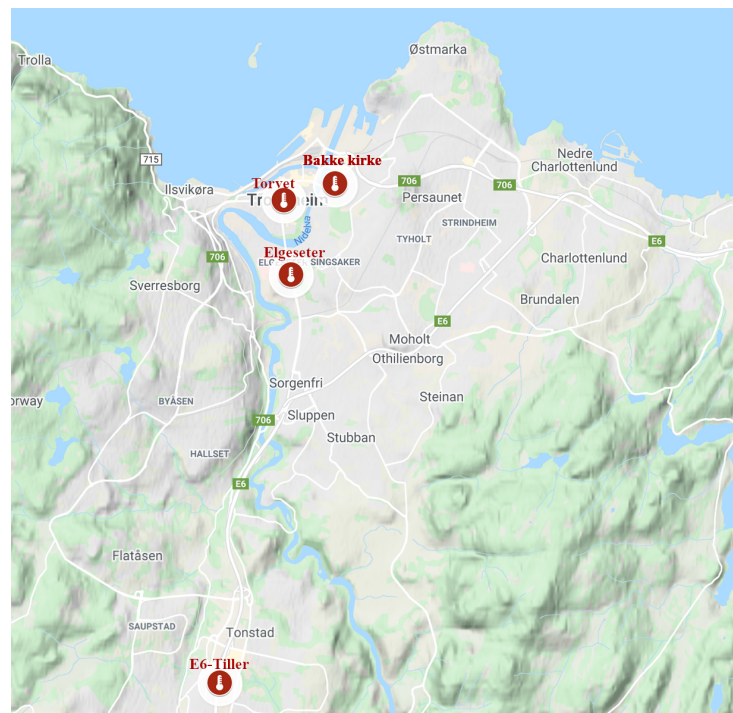


Figure 2. Air quality monitoring stations in Trondheim, Norway.

Although the CAQI (Table 1) specifies five levels of air pollutants, the air quality in the city of Trondheim is usually at *Very Low* and rarely exceeds the *Low* level. For example, Figure 3 shows the air quality level over one year of a representative monitoring station (Elgeseter). Therefore, instead of a multinomial classification task (with five classes), we transform the problem into a threshold exceedance forecast task in which we try to predict the points in time where the air quality exceeds the *Very Low* level. Appendix C contains more details on class-imbalance of air quality due to asymmetry or right-skewed distributions.

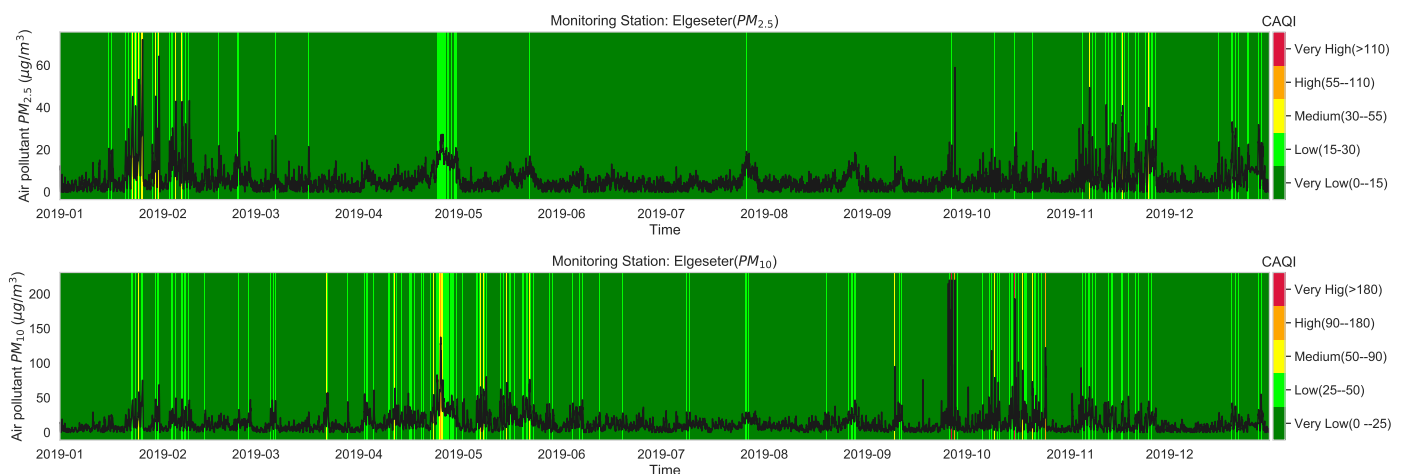


Figure 3. Air quality level over one year of one representative monitoring station in Trondheim, where the air pollutant is commonly at a *Very Low* level and rarely exceeds *Low*.

The air quality dataset we use is a part of the official measuring network in Europe. Specifically, we use the open database of air quality measurements offered by the Norwegian Institute for Air Research (NILU) [37]. The meteorological data are based on historical weather and climate data offered by the Norwegian Meteorological Institute [38]. The traffic data are based on aggregated traffic volumes offered by the Norwegian Public Roads

Administration [39]. A more detailed description of the used datasets can be found in Appendix A.

3.2. Epistemic and Aleatoric Uncertainty

Before quantifying the predictive uncertainty, it is worth distinguishing the different sources of uncertainty and the appropriate actions to reduce them. The first source is model or epistemic uncertainty, which is uncertainty in the model parameters in regions of the input space with little data (i.e., data sparsity). This type of uncertainty can be reduced given enough data. By estimating the epistemic uncertainty of a model, we can obtain its confidence interval (CI). The second source of uncertainty is data or aleatoric uncertainty. It is essentially a noise inherent in the observations (i.e., input-dependent) due to either sensor noise or entropy in the true data generating process. By estimating the aleatoric and epistemic uncertainties, we can obtain the prediction interval (PI) [9,40]. Accordingly, prediction intervals are wider than confidence intervals. The third source of uncertainty is model misspecification, i.e., uncertainty about the general structure of the model, such as model type, number of nodes, number of layers. It is also related to the bias-variance tradeoff [41].

3.3. Non-Probabilistic Baselines

As a baseline for comparison and to motivate the use of evaluation metrics, we test the performance of non-probabilistic models, such as persistence, XGBoost, and gradient boosting models. A simple baseline predictor is a persistence forecast, which uses the diurnal patterns of the observations. To predict a value in the future, we use the value observed 24 h earlier. Figure 4 shows the results of the persistence model when forecasting the PM-value over one month (January 2020). Suppose $\hat{y}_t \in \mathbb{R}$ is the forecast value, while $y_t \in \mathbb{R}$ is the true observed value at time t , then we can evaluate the aggregated accuracy over a time period T using the root-mean-square error:

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2}. \quad (1)$$

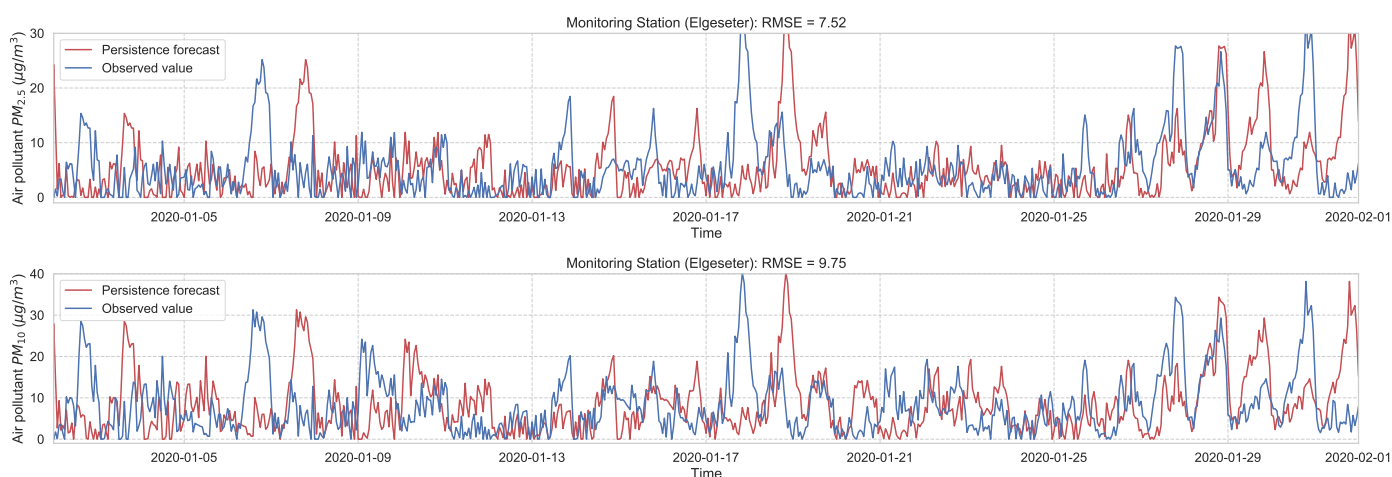


Figure 4. Persistence forecast of air pollutant over one month in one representative monitoring station.

Next we evaluate the performance of XGBoost (*eXtreme Gradient Boosting*) [42] as a non-probabilistic baseline in our problem setting. XGBoost uses the same model representation and inference as Random forests (i.e., gradient-boosted decision trees) but has a different training mechanism since it uses the second-order approximation of the training objective. We train the model over one year of data (2019) and test its performance over one month (January 2020) of a 24-h forecast horizon. Figure 5 shows the results of the XGBoost

model in one representative monitoring station. The results show an improved prediction accuracy compared to the persistence forecast, which illustrates the value of a learned predictor for time-series forecasting of air quality.

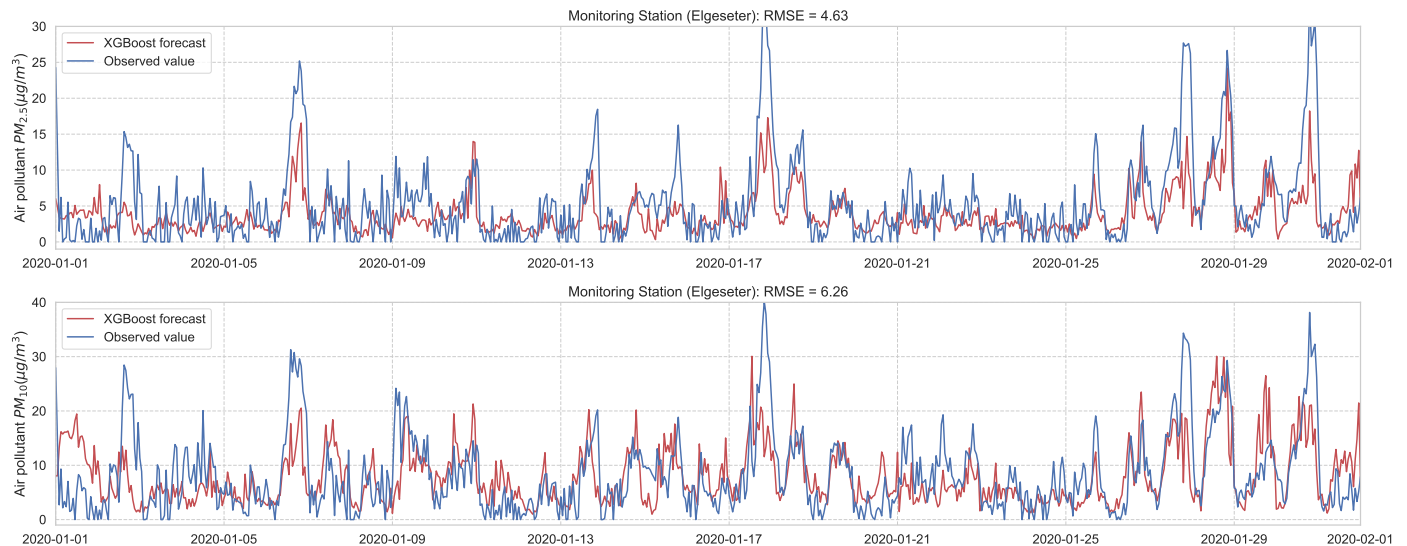


Figure 5. PM-value regression using the XGBoost model over one month in one representative monitoring station.

One advantage of using an XGBoost model is the feasibility of retrieving the feature importance by assigning a score to each input feature, which indicates how useful that feature is when making a prediction, thus contributing to prediction interpretation. Figure 6 shows the feature importance of the XGBoost forecast shown in Figure 5 (a more detailed description of the features can be found in Appendix A).

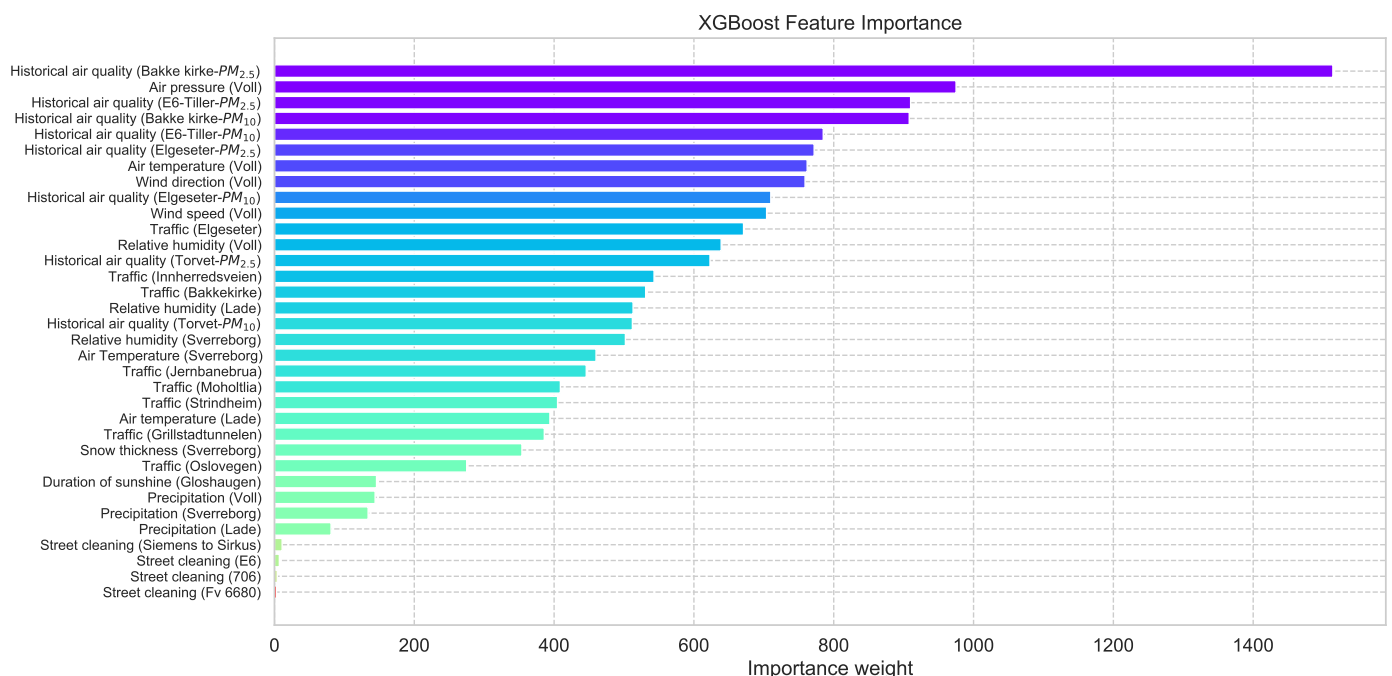


Figure 6. Feature importance of a trained XGBoost model indicating how useful each input feature when making a prediction.

For the threshold exceedance prediction task, we can use XGBoost to train a binary classifier to predict the probability of threshold exceedance. By using a proper scoring rule [43], such as the cross-entropy as a training criterion, we obtain predictive probabilities. Notably, these probabilities do not represent the model (epistemic) uncertainty. Rather, they

represent data (aleatoric) uncertainty. In addition to cross-entropy, we need to evaluate performance using another metric that the model was not optimizing. We can use the Brier score [44], which is a strictly proper scoring rule [43] and metric to evaluate the reliability of predictive probabilities. Suppose at a time, t , the true class label is represented by o_t (1 when pollutant level exceeds the threshold, 0 if not), while the predicted probability is represented by $\hat{p}_t \in [0, 1]$. Then we calculate the cross-entropy (CE) and Brier Score (BS) as follows:

$$CE = -\frac{1}{T} \sum_{t=1}^T (o_t \times \log(\hat{p}_t)) \quad (2)$$

$$BS = \frac{1}{T} \sum_{t=1}^T (o_t - \hat{p}_t)^2 \quad (3)$$

Both metrics heavily punish overconfident, incorrect predictions. Cross-entropy uses exponential punishment (heavily emphasizes tail probabilities) since it is a negative log-likelihood loss. Thus, it is sensitive to the predicted probabilities of the infrequent class (i.e., threshold exceedance events). In contrast, the Brier score uses quadratic punishment since it is a mean square error in the probability space. Thus, it treats the predicted probabilities of both classes equally.

Additionally, we can use the commonly used metrics for classification tasks [45] to evaluate the performance of a deterministic prediction. By converting the predictive probabilities into predictive class labels, we can calculate the true-positive rate tp , the false-positive rate fp , and the false-negative rate fn . Then, we can use the metrics of $Precision : \mathbb{R} \rightarrow [0, 1]$, $Recall : \mathbb{R} \rightarrow [0, 1]$, and $F1 : \mathbb{R} \rightarrow [0, 1]$ score to evaluate the threshold exceedance prediction as following:

$$Precision = \frac{tp}{tp + fp} \quad (4)$$

$$Recall = \frac{tp}{tp + fn} \quad (5)$$

$$F1 = \frac{tp}{tp + 0.5 \times (fp + fn)} \quad (6)$$

Figure 7 shows the results of the XGBoost model when predicting threshold exceedance probability. The blue dots indicate the points in time (hours) when air pollutants actually exceeded the threshold level. The red line represents the predicted probability in percentages. We see that with higher probability, the model predicts a more likely event of threshold exceedance at that specific time.

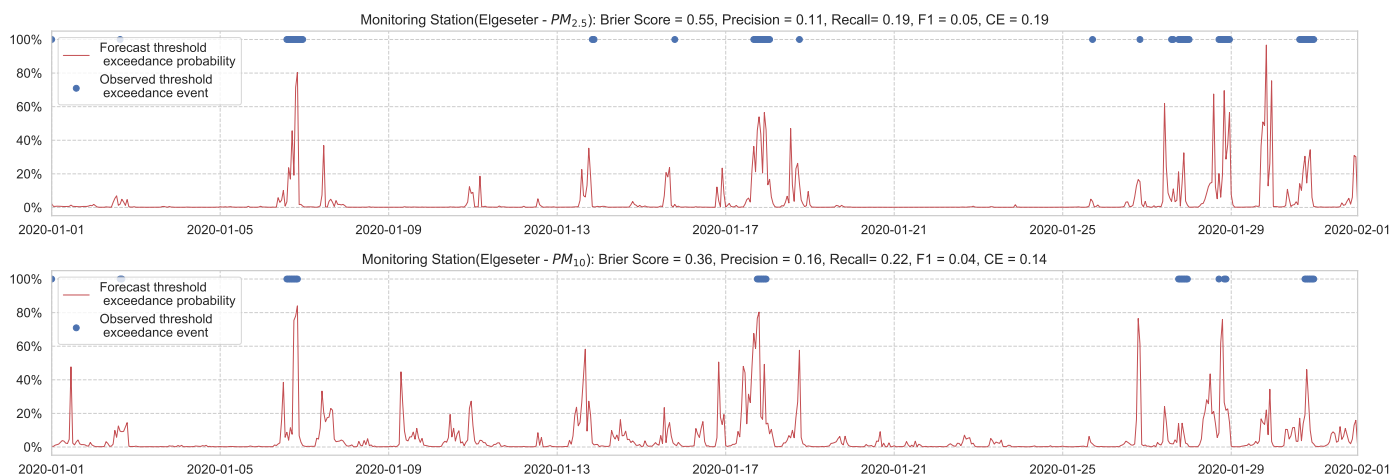


Figure 7. Predicting the threshold exceedance probability of the air pollutant level using an XGBoost model in one representative monitoring station.

3.4. Quantile Regression

Using an XGBoost model leads to an improved prediction accuracy (Figure 5), but with a caveat that it only produces point estimates of the mean. To address this, we can use quantile regression [46] to estimate a specific percentile (i.e., quantile) of the target variable. The quantile regression estimates the conditional quantile by minimizing the absolute residuals. In contrast, regular regression estimates the conditional mean by minimizing the least squared residuals (focuses on central tendency). The advantages of using quantile regression are that it focuses more on dispersion or variability, does not assume homoscedasticity in the target variable (i.e., having the same variance independent of the input variable), and is more robust against outliers. We use a Gradient Tree Boosting model [47] to estimate the 5% and 95% percentiles, as shown in Figure 8.

Given a predicted lower $\hat{L}_t \in \mathbb{R}$ and upper bound $\hat{U}_t \in \mathbb{R}$, we can assess the quality of the generated prediction interval using metrics, such as Prediction Interval Coverage Probability ($PICP : \mathbb{R} \rightarrow [0, 1]$) and Mean Prediction Interval Width ($MPIW : \mathbb{R} \rightarrow [0, \infty)$):

$$PICP = \frac{1}{T} \sum_{t=1}^T \mathbb{1}(y_t - \hat{L}_t) \times \mathbb{1}(\hat{U}_t - y_t) \quad (7)$$

$$MPIW = \frac{1}{T} \sum_{t=1}^T (\hat{U}_t - \hat{L}_t) \quad (8)$$

where $\mathbb{1}$ is the Heaviside step function. $PICP$ indicates how often the prediction interval captures the true values, ranging from 0 (all outside) to 1 (all inside). Intuitively, we seek a model with a narrow prediction interval (i.e., smaller $MPIW$) while capturing the observed data points (i.e., larger $PICP$). For example, when forecasting $PM_{2.5}$ pollutants in Figure 8, $PICP = 0.61$ indicates that in 61% of the time-steps, the observed values are inside the predicted intervals, as compared to 72% when forecasting PM_{10} . In contrast, the predicted intervals are smaller ($MPIW = 7.49$) when forecasting $PM_{2.5}$ as compared to wider predicted intervals ($MPIW = 14.67$) when forecasting PM_{10} .

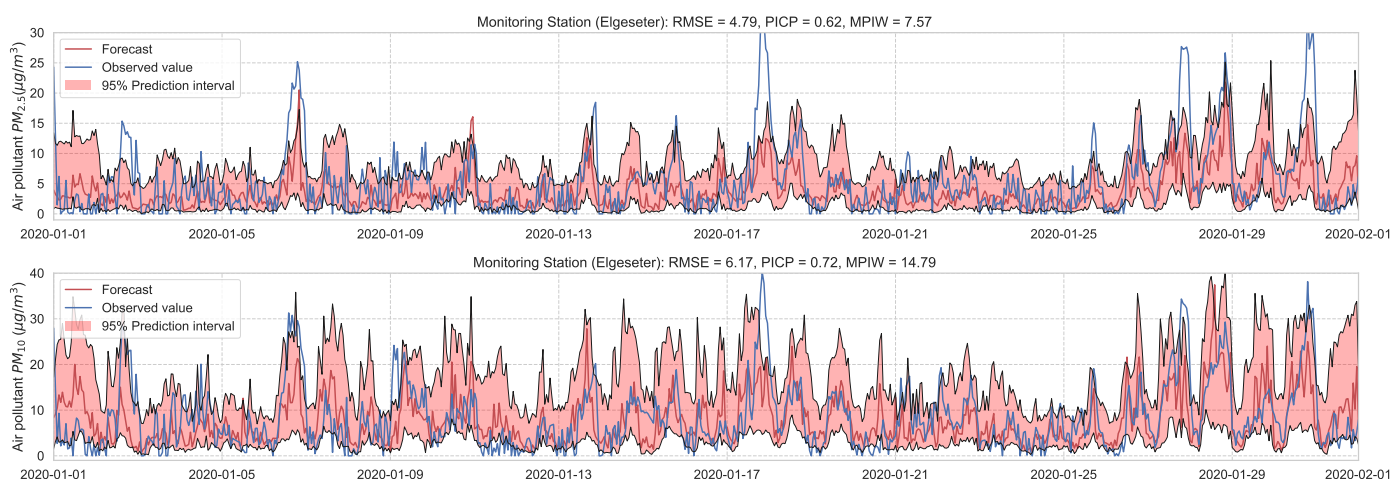


Figure 8. Air quality prediction interval using quantile regression of a Gradient Tree Boosting model.

4. Deep Probabilistic Forecast

This section explores probabilistic models for air quality forecasting and describes how to quantify their predictive uncertainty. We assume we have a training dataset $\mathcal{D} = \{\mathbf{x}_t, y_t\}_{t=1}^T$ with an input feature $\mathbf{x}_t \in \mathbb{R}^D$ and a corresponding observation $y_t \in \mathbb{R}$ for each monitoring station. For the time-series forecasting task, we estimate the aleatoric uncertainty by employing the Mean-Variance Estimation method [48], in which we have a neural network with two output nodes (for every individual time-series) that estimates the mean $\hat{\mu}_t \in \mathbb{R}$ and variance $\hat{\sigma}_t^2 \in (0, \infty)$ of the target probability distribution. Additionally,

we use the negative log-likelihood ($NLL : (0, \infty) \rightarrow (0, \infty)$) as a training criterion since the RMSE does not capture the predictive uncertainty. By treating the observed value y_t as a sample from the target distribution, we can calculate NLL as follows:

$$NLL = 0.5 \left(\log 2\pi\hat{\sigma}_t^2 + \frac{(y_t - \hat{\mu}_t)^2}{\hat{\sigma}_t^2} \right) \quad (9)$$

Notably, we treat the target distributions as having heteroscedastic uncertainty (i.e., non-constant variance) and diagonal covariance matrices, which simplifies the training criterion of multivariate time-series. For the threshold exceedance prediction task, we estimate the aleatoric uncertainty by having a neural network that outputs the predictive probability in terms of cross-entropy.

Although epistemic uncertainty is model-dependent, it is essentially estimated by measuring the dispersion in predictions when running several inference steps over a specific data point. For the PM-value regression task, every single forward pass (i) outputs a normal distribution (with mean $\hat{\mu}_{t,i}$ and variance $\hat{\sigma}_{t,i}^2$). Thus, multiple forward passes result in a (uniformly weighted) mixture of normal distributions with the mean $\hat{\mu}_{t,mix} \in \mathbb{R}$ and variance $\hat{\sigma}_{t,mix}^2 \in (0, \infty)$ calculated as follows:

$$\hat{\mu}_{t,mix} = \frac{1}{M} \sum_{i=1}^M \hat{\mu}_{t,i} \quad (10)$$

$$\hat{\sigma}_{t,mix}^2 = \frac{1}{M} \sum_{i=1}^M (\hat{\sigma}_{t,i}^2 + \hat{\mu}_{t,i}^2) - \hat{\mu}_{t,mix}^2 \quad (11)$$

Given a predicted mean and variance, we can then estimate a point prediction $\hat{y}_t = \hat{\mu}_{t,mix}$, a lower bound $\hat{L}_t = \hat{\mu}_{t,mix} - z\hat{\sigma}_{t,mix}$, and an upper bound $\hat{U}_t = \hat{\mu}_{t,mix} + z\hat{\sigma}_{t,mix}$, where z is the standard score of a normal distribution. For example, with 95% prediction interval (i.e., $P(L_t < y_t < U_t) = 0.95$), we use $z = 1.96$. For the threshold exceedance prediction task, every single forward pass outputs a predictive probability. Thus, we can combine predictions from multiple forward passes by simply averaging the predicted probabilities.

For empirical evaluation, we use the NLL and cross-entropy as evaluation metrics. Additionally, we use the Continuous Ranked Probability Score (CRPS) [43]. It is a widely used metric to evaluate probabilistic forecasts that generalizes the MAE to a probabilistic setting. CRPS measures the difference in the cumulative distribution function (CDF) between the forecast and the true observation. It can be derived analytically for parametric distributions or estimated using samples if the CDF is unknown (e.g., originating from VI or MCMC). Given a forecast CDF \mathbf{F} and an empirical CDF of scalar observation y :

$$CRPS(\mathbf{F}, y) = \int_{\mathbb{R}} (\mathbf{F}(\hat{y}) - \mathbb{1}(\hat{y} - y))^2 d\hat{y} \quad (12)$$

4.1. Bayesian Neural Networks (BNNs)

In BNNs, we use Bayesian methods for inferring a posterior distribution over the weights $p(\mathbf{w})$ rather than being constrained to weights of fixed values [2]. These weight distributions are parameterized by trainable variables θ . For example, the trainable variable can represent the mean and variance of a Gaussian distribution $\theta = (\mu, \sigma^2)$, from which the weights can be sampled $\mathbf{w} \sim \mathcal{N}(\mu, \sigma^2)$.

The objective of training is to calculate the posterior, but obtaining explicit posterior densities through Bayesian inference is intractable. Additionally, using Markov chain Monte Carlo (MCMC) to estimate the posterior can be computationally prohibitive. Instead, we can leverage new techniques in variational inference [49,50] that make BNNs computationally feasible by using a variational approximation $q(\mathbf{w}|\theta)$ to the posterior. Thereby, the goal of learning is to find the variational parameters that minimize the Kullback–Leibler (KL) divergence between the variational approximation $q(\mathbf{w}|\theta)$ and the true

posterior distribution given training data $p(\mathbf{w}|\mathcal{D})$. This can be achieved by minimizing the negative variational lower bound of the marginal likelihood [30]:

$$Loss(\theta) = KL[q(\mathbf{w}|\theta)||p(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)} [\log p(\mathcal{D}|\mathbf{w})] \quad (13)$$

Assuming Gaussian prior and posterior, we can compute the KL divergence in a closed-form:

$$KL[q(\mathbf{w}|\theta)||p(\mathbf{w})] = \log \frac{\sigma_p}{\sigma_q} + \frac{\sigma_q^2 + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^2}{2\sigma_p^2} - 0.5 \quad (14)$$

Additionally, the distribution over activations will also be Gaussian. Thus, we can take advantage of the local reparameterization trick [51], in which we sample from the distribution over activations rather than sampling the weights individually. Consequently, we reduce the variance of stochastic gradients, resulting in faster and more stable training. However, to allow for more flexibility and adaptations to a wide range of situations, we can use non-Gaussian distributions over the weights and MC gradients, as proposed by [3]:

$$Loss(\theta) \approx \sum_{i=1}^M \log q(\mathbf{w}^i|\theta) - \log P(\mathbf{w}^i) - \log P(\mathcal{D}|\mathbf{w}^i) \quad (15)$$

where $-\log P(\mathcal{D}|\mathbf{w}^i)$ is the NLL for time-series forecasting Equation (9) or the cross-entropy for the threshold exceedance prediction Equation (2).

Our implementation uses Laplace distributions as priors and Gaussian as approximate posteriors in time-series forecasting, resulting in better empirical performance. For threshold exceedance prediction, using Gaussian for both the priors and posteriors and using the local reparameterization trick results in better performance. During inference, we sample the weight distributions and perform a forward pass to obtain a prediction. We use ($M = 1000$) samples for every data point to estimate the model's uncertainty. This corresponds to sampling from infinite ensembles of neural networks. Therefore, combining the outputs from different samples gives information on the model's uncertainty.

We train the BNN model using training data of one year (2019). Figure 9 shows the learning curves for the PM-value regression task. Then we evaluate the model using data of one month (January 2020). Figures 10 and 11 show the results of PM-value regression and threshold exceedance classification in one representative monitoring station. Table 2 shows a summary of performance results in all monitoring stations. The arrows alongside the metrics indicate which direction is better for that specific metric. Additionally, Appendix D.1 shows the performance results in all monitoring stations.

Table 2. Summary of performance results when forecasting the PM-value and threshold exceedance using a BNNs model.

Station	Particulate	PM-Value Regression				Threshold Exceedance Classification					
		RMSE↓	PICP↑	MPIW↓	CRPS↓	NLL↓	Brier↓	Precision↑	Recall↑	F1↑	CE↓
Bakke Kirke	$PM_{2.5}$	4.81	0.99	17.62	0.51	1.29	0.04	1.00	0.44	0.61	0.13
	PM_{10}	5.86	0.94	26.12	0.50	1.28	0.03	1.00	0.30	0.47	0.09
E6-Tiller	$PM_{2.5}$	3.77	0.92	13.25	0.54	1.39	0.02	0.00	0.00	0.00	0.08
	PM_{10}	9.40	0.92	34.18	0.48	1.26	0.06	0.00	0.00	0.00	0.23
Elgeseter	$PM_{2.5}$	3.93	0.91	12.79	0.53	1.36	0.03	0.88	0.42	0.56	0.12
	PM_{10}	5.17	0.90	25.07	0.47	1.28	0.03	0.55	0.19	0.29	0.12
Torvet	$PM_{2.5}$	4.07	0.90	10.83	0.48	1.30	0.03	0.75	0.46	0.57	0.13
	PM_{10}	5.25	0.93	18.47	0.43	1.17	0.03	0.50	0.23	0.32	0.10

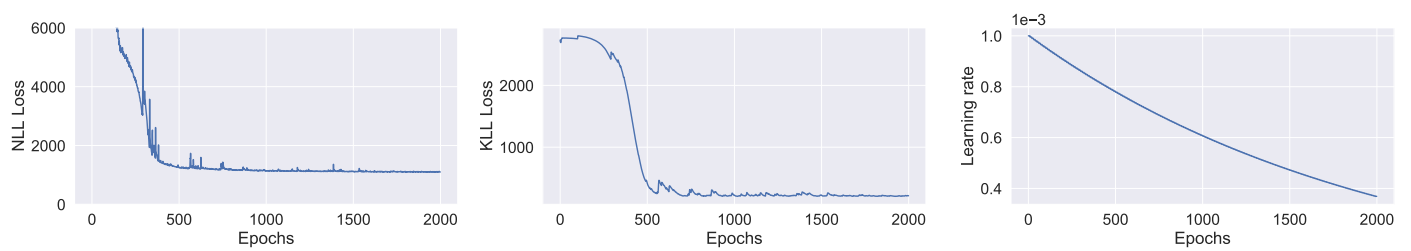


Figure 9. Learning curve of training a BNN model to forecast PM-values. (Left): negative log-likelihood loss; (Center): KL loss estimated using MC sampling; (Right): learning rate of exponential decay.

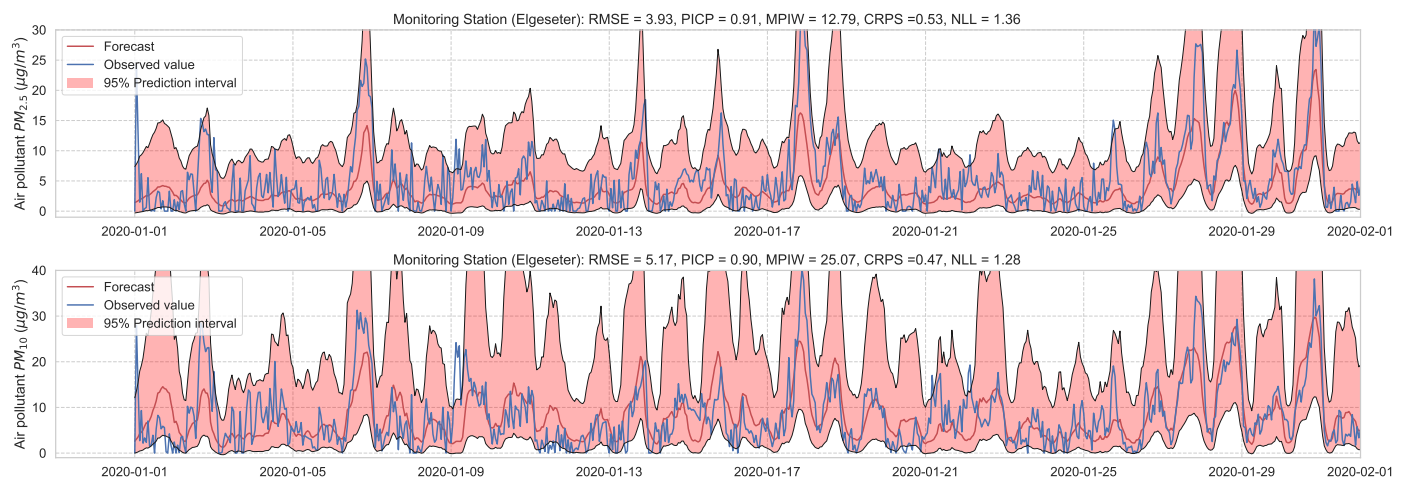


Figure 10. Probabilistic forecasting of multivariate time-series of air quality using a BNN model in one representative monitoring station.

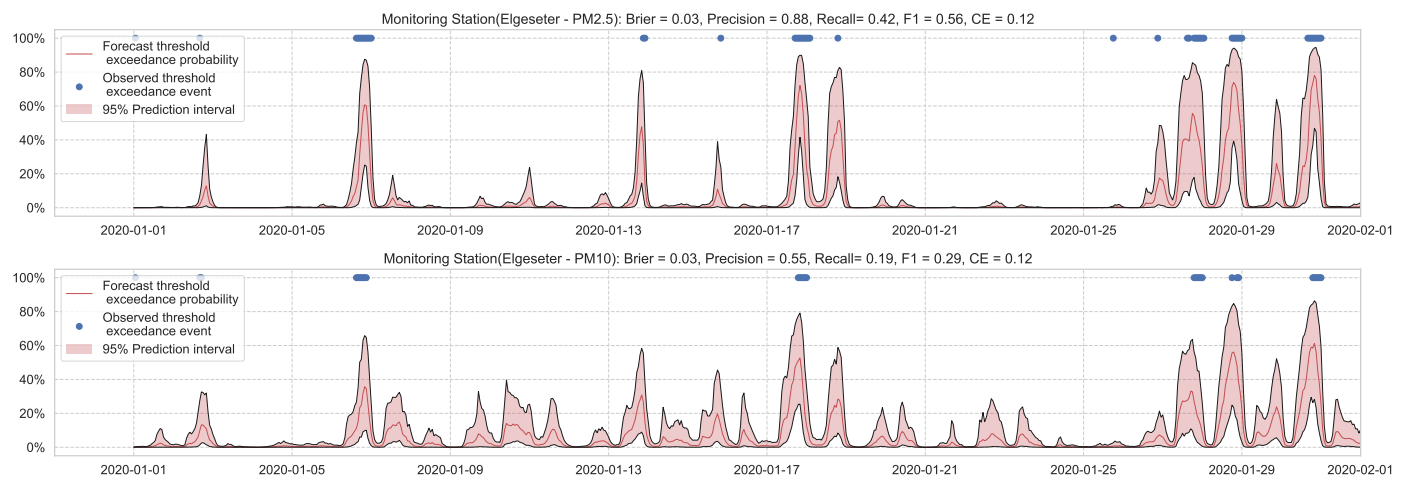


Figure 11. Predicting threshold exceedance probability of air pollutant level using a BNN model.

A less efficient approach to predicting threshold exceedance probability is by transforming the time-series forecast into a binary prediction. Figure 12 shows the results of a BNN model. The result demonstrates that transforming PM-value regression into threshold exceedance classification leads to a loss in performance.

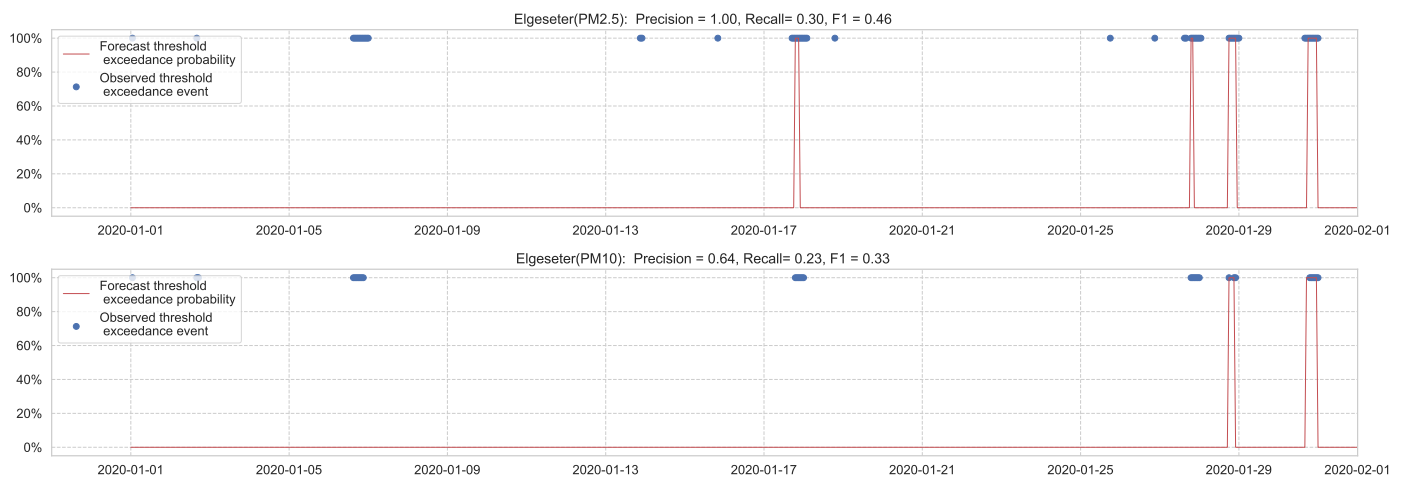


Figure 12. Predicting threshold exceedance probability by transforming PM-value regression into binary predictions.

4.2. Standard Neural Networks with MC Dropout

Although BNNs are more flexible in reasoning about the model's uncertainty with Bayesian analysis, they are computationally less efficient and take longer to converge than standard (non-Bayesian) neural networks. Additionally, they require double the number of parameters at deployment compared to neural networks of the same size. A possible solution is to gracefully prune the weights with the lowest signal-to-noise ratio [3], but this leads to a loss in uncertainty information. Therefore, it would be more convenient to obtain uncertainty directly from standard neural networks. One simple approach we can use is Monte Carlo dropout as an approximate Bayesian method for representing model uncertainty. As shown in [4], MC dropout can be interpreted as performing variational inference. More specifically, it is mathematically equivalent to an approximation of a probabilistic deep Gaussian process.

Essentially, we train a standard neural network model with dropout. Then we keep the dropout during inference and run multiple forward passes using the same data input. This corresponds to sampling nodes, which is equivalent to sampling from ensembles of neural networks [52]. By measuring the spread in predictions, we estimate the predictive uncertainty. In our implementations, we train a standard neural network model with a 50% dropout rate. Then we evaluate it with 50% dropout and ($M = 1000$) samples. Figures 13 and 14 show the results of the PM-value regression and threshold exceedance classification in one representative monitoring station. Table 3 shows a summary of performance results in all monitoring stations, while Appendix D.2 contains the corresponding figures.

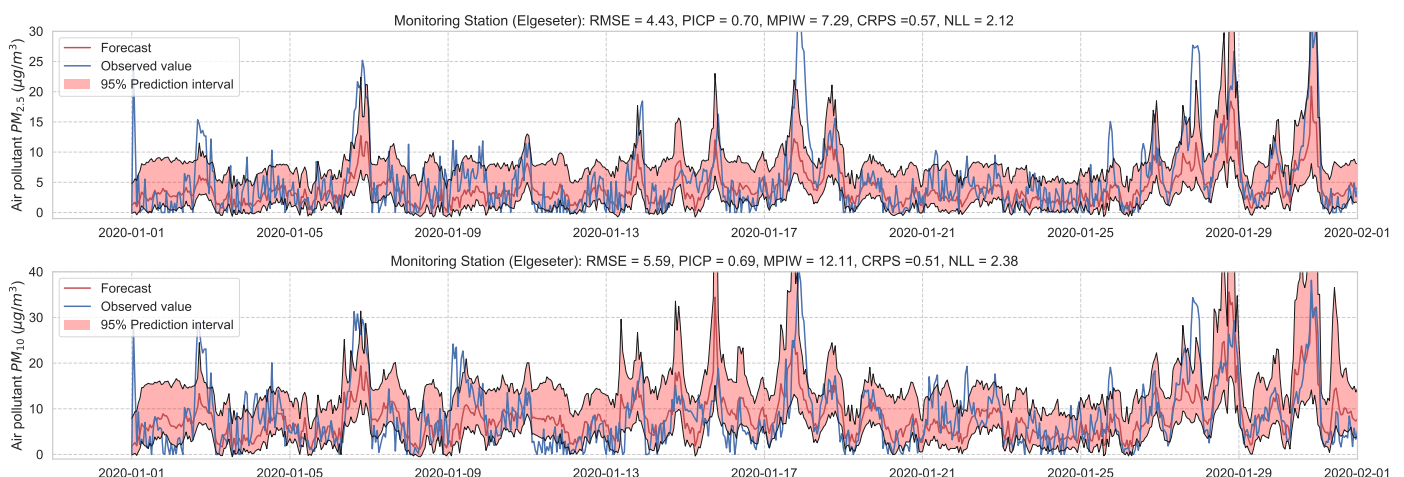


Figure 13. Probabilistic forecasting of multivariate time-series air quality using a standard neural network model with MC dropout.

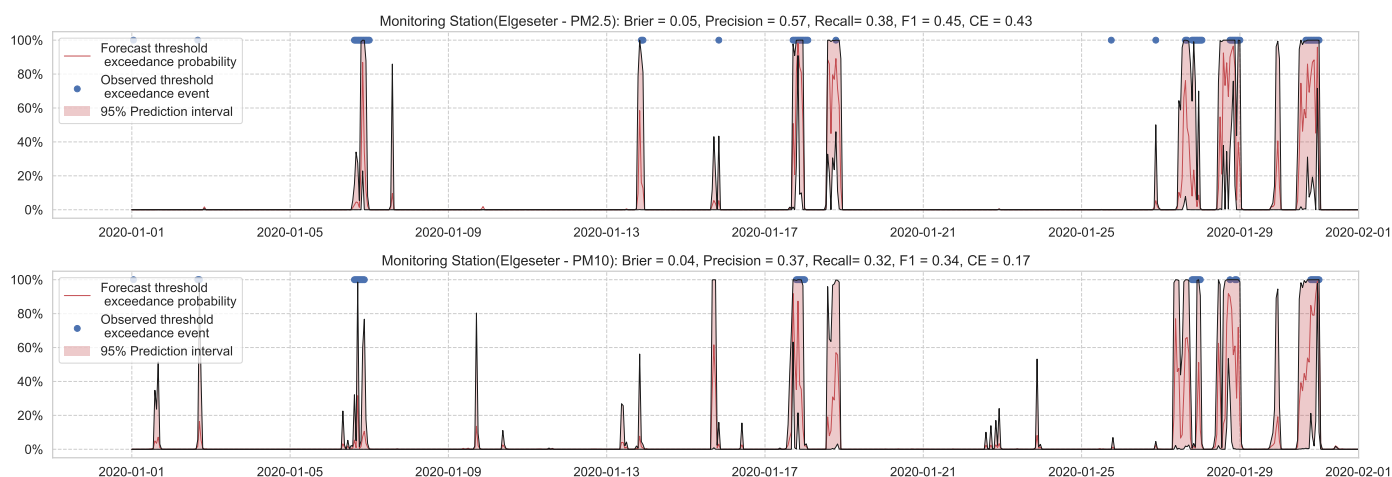


Figure 14. Predicting the threshold exceedance probability of air pollutants level using a standard neural network with MC dropout.

Table 3. Summary of performance results when forecasting the PM-value and threshold exceedance using a standard neural network with MC dropout.

Station	Particulate	PM-Value Regression					Threshold Exceedance Classification					
		RMSE↓	PICP↑	MPIW↓	CRPS↓	NLL↓	Brier↓	Precision ↑	Recall↑	F1↑	CE↓	
Bakke kirke	PM _{2.5}	5.34	0.69	9.40	0.60	2.30	0.04	0.65	0.51	0.57	0.31	
	PM ₁₀	6.42	0.66	12.45	0.59	3.35	0.03	0.67	0.48	0.56	0.10	
E6-Tiller	PM _{2.5}	3.75	0.72	7.26	0.60	2.24	0.01	0.00	0.00	0.00	0.24	
	PM ₁₀	9.49	0.71	16.62	0.51	2.30	0.07	0.18	0.04	0.06	0.57	
Elgeseter	PM _{2.5}	4.43	0.70	7.29	0.57	2.12	0.05	0.57	0.38	0.45	0.43	
	PM ₁₀	5.59	0.69	12.11	0.51	2.38	0.04	0.37	0.32	0.34	0.17	
Torvet	PM _{2.5}	4.60	0.55	5.26	0.57	2.91	0.04	0.68	0.44	0.53	0.33	
	PM ₁₀	5.63	0.62	8.94	0.51	2.51	0.03	0.56	0.35	0.43	0.14	

4.3. Deep Ensembles

An established method to improve performance is to train an ensemble of neural networks with different configurations [53]. Additionally, many model types can be interpreted by using ensemble methods. For example, sampling weights of BNNs is equivalent to sampling from an infinite ensemble of networks, while sampling the nodes in MC dropout compares to sampling from a finite ensemble [52]. Remarkably, we can also use (deep) ensembles to estimate the predictive uncertainty in neural network models [5].

Essentially, we train multiple neural networks with different parameter initialization on the same data. The stochastic optimization and random initialization ensure that the trained networks are sufficiently independent. During inference, we run a forward pass on the multiple neural networks using the same data input. We estimate the uncertainty by measuring the dispersion in predictions resulting from the multiple neural networks. In our implementation, we train an ensemble of 10 networks. Figures 15 and 16 show the results of PM-value regression and threshold exceedance classification in one representative monitoring station. Table 4 shows a summary of performance results in all monitoring stations, while Appendix D.3 contains the corresponding figures.

The main drawback of the deep ensemble method is the computational cost and number of parameters required at run time compared to standard neural networks, which is an order of magnitude in our case. One solution is to use knowledge distillation [54] to compress the knowledge from the ensemble into a single model, but this also leads to a loss in uncertainty information.

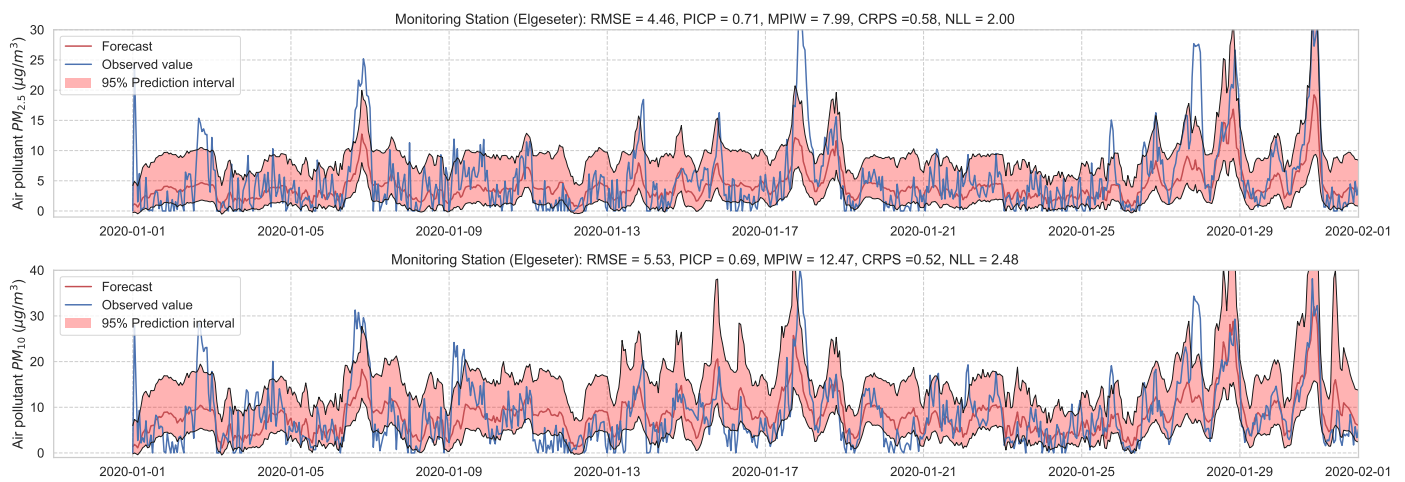


Figure 15. Probabilistic forecasting of multivariate time-series air quality using a deep ensemble.

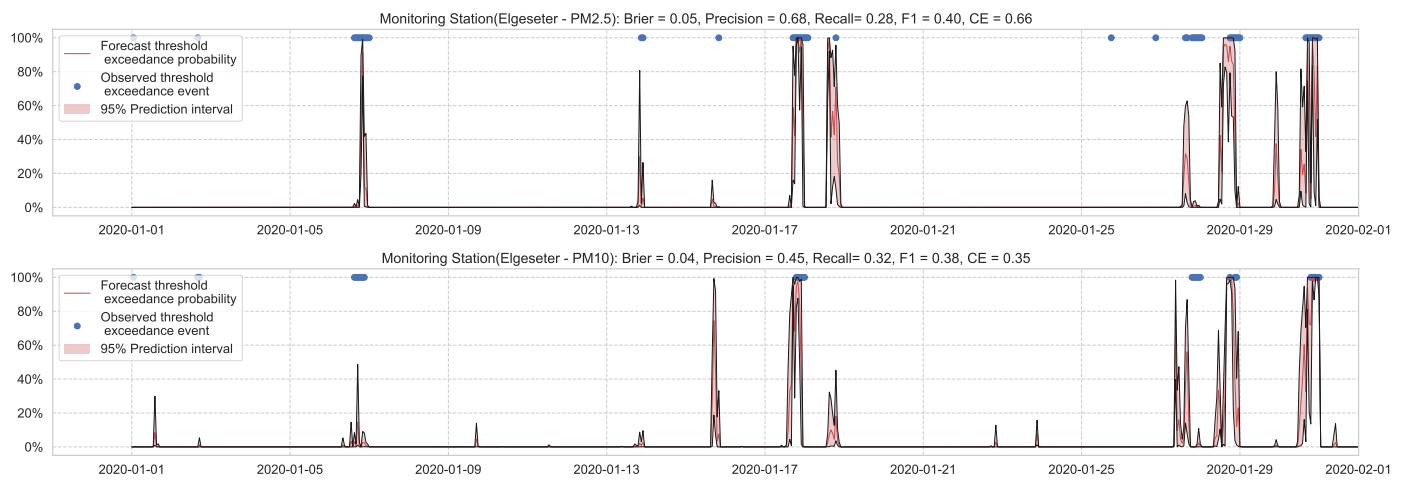


Figure 16. Predicting threshold exceedance probability of air pollutants level using a deep ensemble.

Table 4. Summary of performance results when forecasting the PM-value and threshold exceedance using a deep ensemble.

Station	Particulate	PM-Value Regression					Threshold Exceedance Classification				
		RMSE↓	PICIP↑	MPIW↓	CRPS↓	NLL↓	Brier↓	Precision ↑	Recall↑	F1↑	CE↓
Bakke kirke	PM _{2.5}	5.29	0.77	11.65	0.57	1.67	0.05	0.69	0.53	0.60	0.55
	PM ₁₀	6.21	0.70	14.00	0.57	2.46	0.03	0.60	0.36	0.45	0.26
E6-Tiller	PM _{2.5}	3.78	0.77	8.46	0.58	1.84	0.01	0.00	0.00	0.00	0.34
	PM ₁₀	9.44	0.72	16.07	0.50	2.14	0.07	0.31	0.08	0.12	1.16
Elgeseter	PM _{2.5}	4.46	0.71	7.99	0.58	2.00	0.05	0.68	0.28	0.40	0.66
	PM ₁₀	5.53	0.69	12.47	0.52	2.48	0.04	0.45	0.32	0.38	0.35
Torvet	PM _{2.5}	4.45	0.57	5.13	0.56	2.66	0.04	0.73	0.31	0.43	0.55
	PM ₁₀	5.39	0.64	8.68	0.49	2.19	0.03	0.62	0.19	0.29	0.30

4.4. Recurrent Neural Network with MC Dropout

While standard neural networks are powerful models at representational learning, they do not exploit the inherent temporal correlation in air quality data since they act only on static, fixed contextual windows. To address this shortcoming, we can use recurrent neural networks (RNNs), which have cyclic connections from previous time steps, to learn the temporal dynamics of sequential data. Specifically, a hidden state from the last time step is stored and used in addition to the input to generate the current state and output.

One class of RNNs is the long short-term memory (LSTM), which is used extensively in sequence modeling tasks, such as modeling language [55], forecasting weather [56] and traffic [57], recognizing human activity [58], and recently forecasting COVID-19 transmission [59]. An LSTM has gated memory cells to control how much information to forget from previous states and how much information to use to update current states [60].

A simple approach to capture uncertainty within an LSTM model is to use Monte Carlo dropout to approximate Bayesian inference [6]. To implement dropout, we apply a mask that randomly drops some network units with their inputs, output and recurrent connections [61]. Our implementation trains a model of two LSTM layers with a 50% dropout rate and evaluates it with 50% dropout and ($M = 1000$) samples. Figures 17 and 18 show the results of the PM-value regression and threshold exceedance classification in one representative monitoring station. Table 5 shows a summary of performance results in all monitoring stations, while Appendix D.4 contains the corresponding figures.

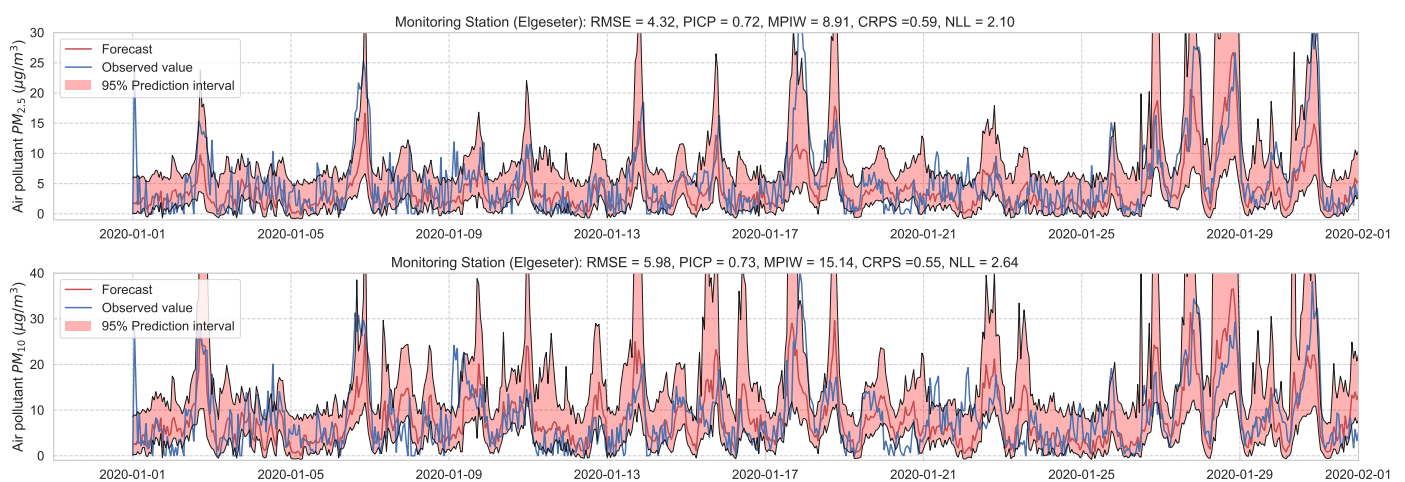


Figure 17. Probabilistic forecasting of multivariate time-series air quality using an LSTM model with MC dropout.

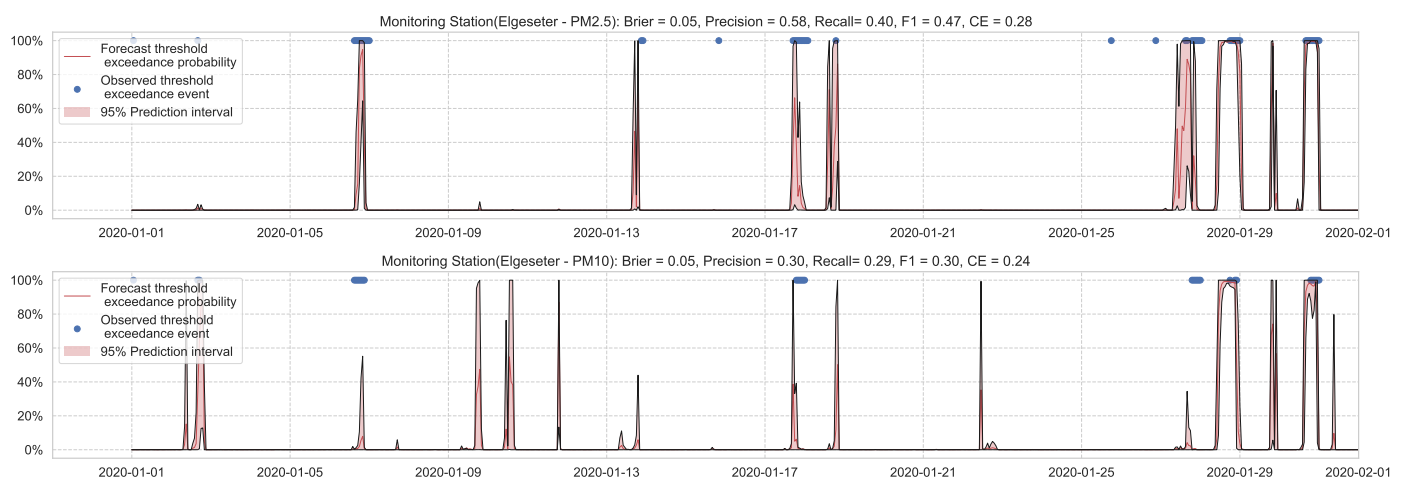


Figure 18. Predicting threshold exceedance probability of air pollutants level using an LSTM model with MC dropout.

Table 5. Summary of performance results when forecasting PM-value or threshold exceedance using an LSTM model with MC dropout.

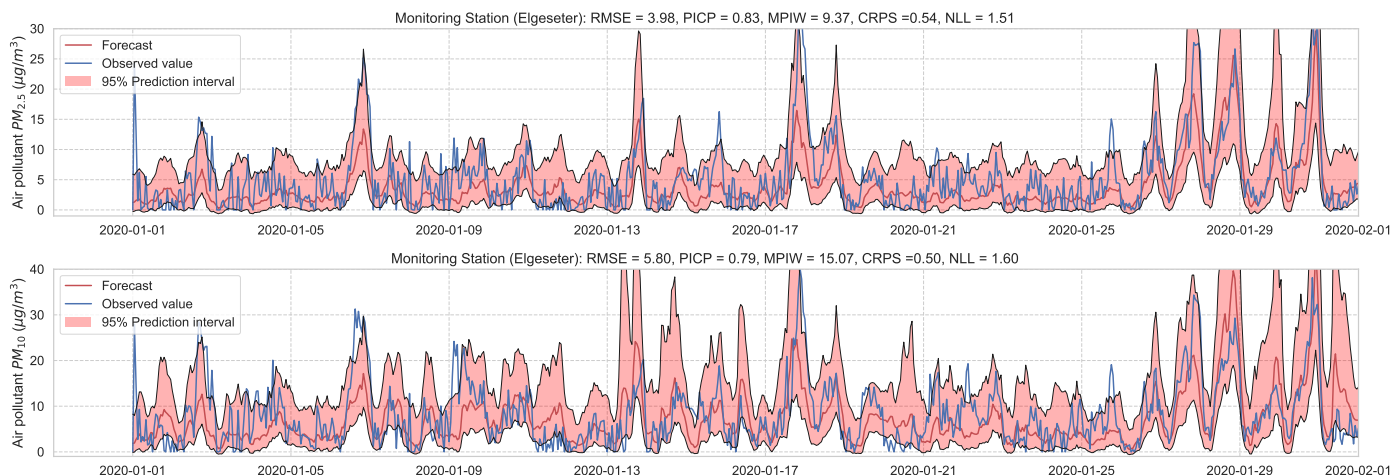
Station	Particulate	PM-Value Regression					Threshold Exceedance Classification				
		RMSE↓	PICP↑	MPIW↓	CRPS↓	NLL↓	Brier↓	Precision ↑	Recall↑	F1↑	CE↓
Bakke kirke	PM _{2.5}	5.01	0.88	14.01	0.53	1.47	0.05	0.66	0.53	0.58	0.25
	PM ₁₀	6.25	0.82	19.28	0.54	1.78	0.03	0.59	0.48	0.53	0.14
E6-Tiller	PM _{2.5}	3.90	0.72	7.45	0.62	2.31	0.02	0.00	0.00	0.00	0.11
	PM ₁₀	9.68	0.74	18.96	0.53	2.03	0.08	0.24	0.12	0.16	0.43
Elgeseter	PM _{2.5}	4.32	0.72	8.91	0.59	2.10	0.05	0.58	0.40	0.47	0.28
	PM ₁₀	5.98	0.73	15.14	0.55	2.64	0.05	0.30	0.29	0.30	0.24
Torvet	PM _{2.5}	4.19	0.56	6.88	0.58	4.79	0.05	0.58	0.42	0.49	0.30
	PM ₁₀	5.81	0.61	11.33	0.54	4.03	0.03	0.43	0.35	0.38	0.1

4.5. Graph Neural Networks with MC Dropout

By using RNNs, we can capture the temporal (i.e., intra-series) correlations in the time-series of air quality data. However, we need also to exploit the inherent structural (i.e., inter-series) correlations between multiple sensing stations. We can use Graph Neural Networks (GNNs) to address this, which operate on graph-structured data. In essence, GNNs update the features of a graph node by aggregating the features of its adjacent nodes. For example, a node can be a single monitoring station in our case. In the end, GNNs apply a shared layer on each node to obtain a prediction for each node.

In our setting, we assume each sensing station to be a node in a weighted and directed graph, represented by a learnable adjacency matrix. We use a slight variation of the GNNs suggested by Cao et al. [62] to forecast multivariate time-series of air quality. The main idea is to learn a correlation graph directly from data (i.e., learn a graph of sensor nodes without a pre-defined typology) and then learn the structural and temporal correlation in the frequency domain. To capture the temporal correlations, we use a layer of 1D convolution, and three sub-layers of Gated Linear Units (GLUs) [63], while we use Graph Convolutional Networks (GCNs) [64] to capture the structural correlations. In the end, we use two shared layers of fully connected neural networks to predict each node's output.

We use the MC dropout applied to the GLUs, GCNs, and to the fully connected layers to estimate model's uncertainty [65]. We train the model with a 50% dropout rate and evaluate it with 50% dropout and ($M = 1000$) samples. Figures 19 and 20 show the results of the PM-value regression and threshold exceedance classification in one representative monitoring station. Table 6 shows a summary of performance results in all monitoring stations, while Appendix D.5 contains more-detailed figures.

**Figure 19.** Probabilistic forecasting of multivariate time-series air quality using a GNN model with MC dropout.

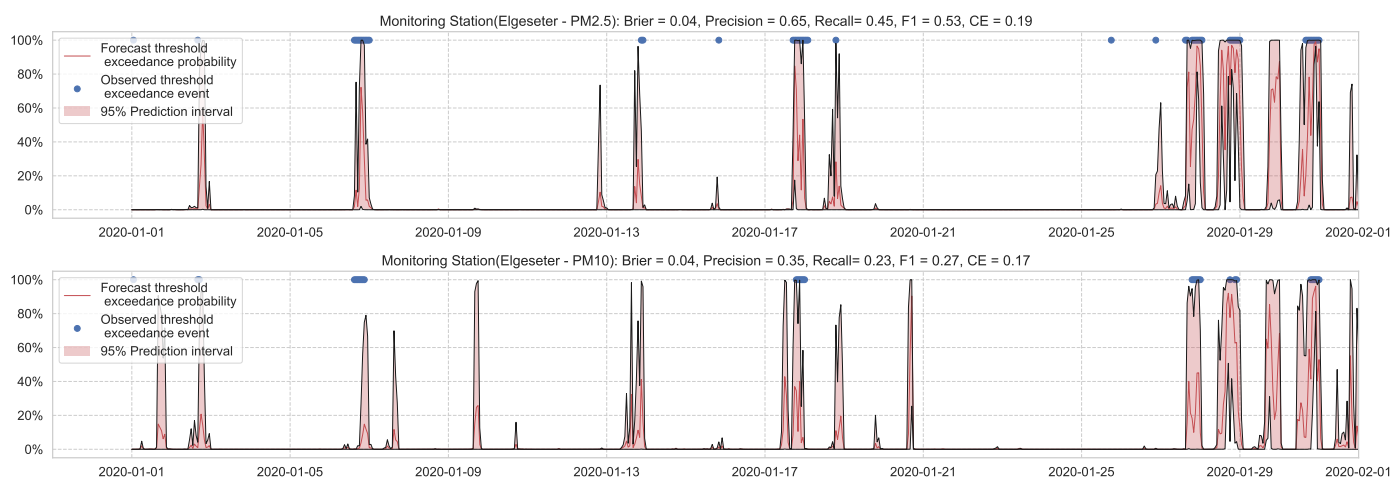


Figure 20. Predicting threshold exceedance probability of air pollutants level using a GNN model with MC dropout.

Table 6. Summary of performance results when forecasting the PM-value or threshold exceedance using a GNN model with MC dropout.

Station	Particulate	PM-Value Regression					Threshold Exceedance Classification				
		RMSE↓	PICP↑	MPIW↓	CRPS↓	NLL↓	Brier↓	Precision ↑	Recall↑	F1↑	CE↓
Bakke kirke	$PM_{2.5}$	4.70	0.88	12.33	0.52	1.41	0.05	0.61	0.53	0.56	0.21
	PM_{10}	6.26	0.79	16.10	0.54	1.83	0.03	0.43	0.36	0.39	0.11
E6-Tiller	$PM_{2.5}$	3.80	0.83	9.14	0.57	1.60	0.02	0.00	0.00	0.00	0.11
	PM_{10}	9.46	0.80	19.89	0.48	1.59	0.07	0.19	0.06	0.09	0.35
Elgeseter	$PM_{2.5}$	3.98	0.83	9.37	0.54	1.51	0.04	0.65	0.45	0.53	0.19
	PM_{10}	5.80	0.79	15.07	0.50	1.60	0.04	0.35	0.23	0.27	0.17
Torvet	$PM_{2.5}$	4.27	0.68	6.19	0.50	2.04	0.05	0.55	0.46	0.50	0.22
	PM_{10}	5.55	0.70	10.39	0.47	1.83	0.03	0.36	0.35	0.35	0.11

4.6. Stochastic Weight Averaging–Gaussian (SWAG)

An alternative approach to estimate uncertainty in a neural network is to approximate (Gaussian) posterior distributions over the weights using the geometric information in the trajectory of a stochastic optimizer. This approach is named Stochastic Weight Averaging–Gaussian (SWAG) [7]. Notably, SWAG does not optimize the approximate distributions directly, such as in BNNs. Instead, it estimates the mean by calculating a running average of the weights traversed by the optimizer with a modified learning rate schedule (stochastic weight averaging [66]). In addition, SWAG estimates the standard deviation by a diagonal covariance plus of a low-rank deviation matrix using information from a running average of the second moment of the traversed weights.

During inference, we run multiple forward passes using the same data input while drawing samples of the weights from the approximate posterior. By measuring the spread in predictions, we estimate the predictive uncertainty. In our implementations, we train a simple feed-forward neural network model with SWAG and evaluate it with ($M = 1000$) samples. Figures 21 and 22 show the results of the PM-value regression and threshold exceedance classification in one representative monitoring station. Table 7 shows a summary of performance results in all monitoring stations, and Appendix D.6 contains the corresponding figures.

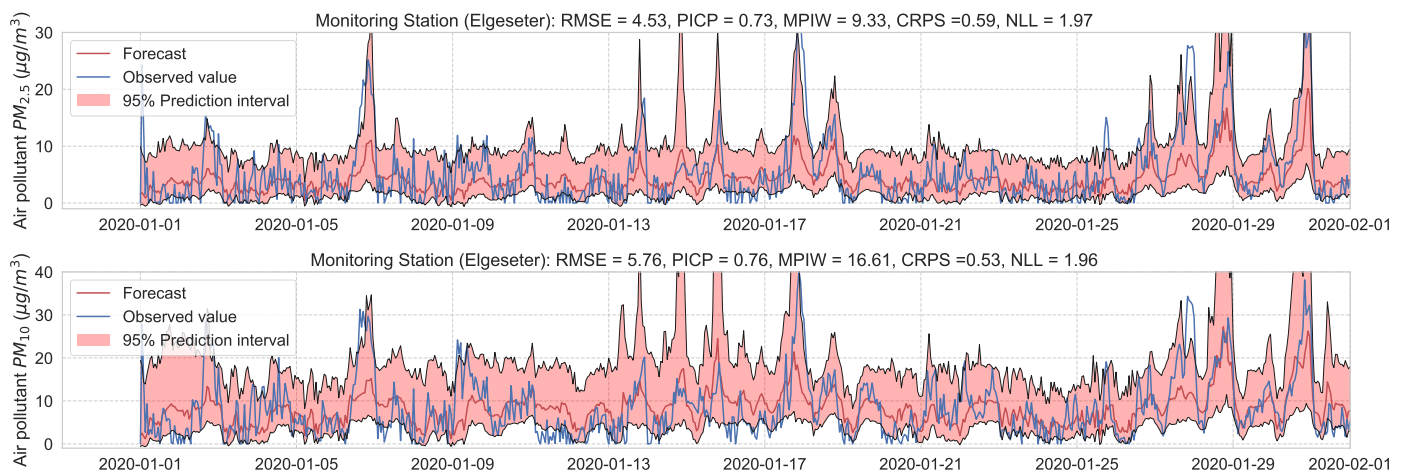


Figure 21. Probabilistic PM-value regression using a SWAG model.

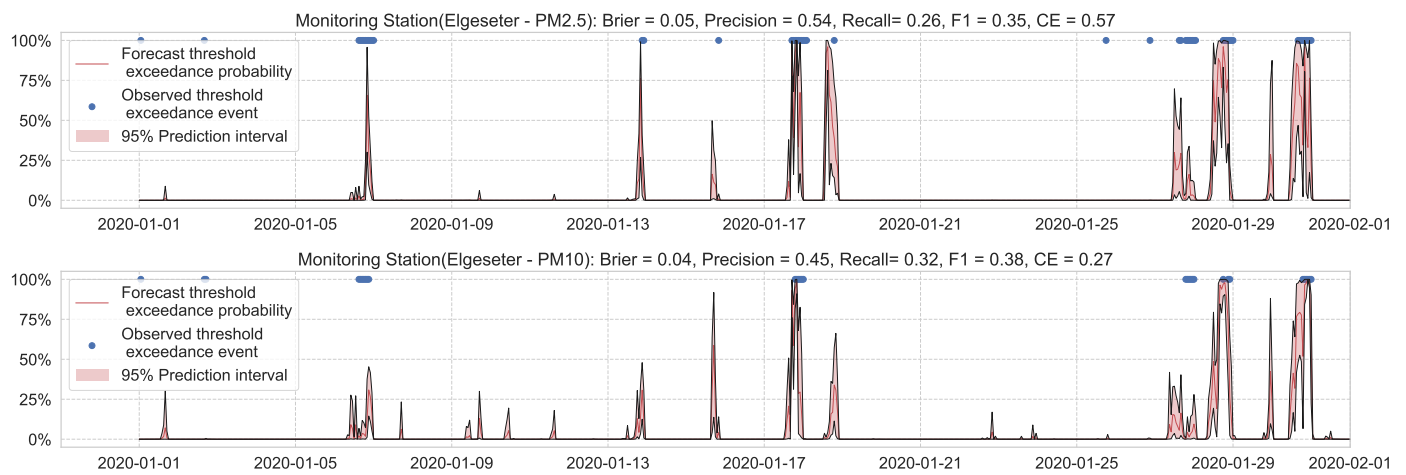


Figure 22. Probabilistic threshold exceedance classification using a SWAG model.

Table 7. Summary of performance results when using a SWAG model with adversarial training.

Station	Particulate	PM-Value Regression					Threshold Exceedance Classification				
		RMSE↓	PICP↑	MPIW↓	CRPS↓	NLL↓	Brier↓	Precision ↑	Recall↑	F1↑	CE↓
Bakke kirke	PM _{2.5}	5.51	0.79	13.13	0.58	1.64	0.04	0.66	0.64	0.65	0.20
	PM ₁₀	6.66	0.78	17.95	0.57	2.03	0.04	0.49	0.61	0.54	0.12
E6-Tiller	PM _{2.5}	3.76	0.79	9.25	0.59	1.82	0.01	0.00	0.00	0.00	0.10
	PM ₁₀	9.35	0.82	21.28	0.49	1.73	0.08	0.19	0.08	0.11	0.49
Elgeseter	PM _{2.5}	4.53	0.73	9.33	0.59	1.97	0.04	0.60	0.45	0.52	0.21
	PM ₁₀	5.76	0.76	16.61	0.53	1.96	0.04	0.37	0.45	0.41	0.18
Torvet	PM _{2.5}	4.58	0.79	10.33	0.54	1.63	0.04	0.67	0.50	0.57	0.20
	PM ₁₀	5.62	0.71	12.48	0.50	1.76	0.03	0.50	0.42	0.46	0.13

4.7. Improving Uncertainty Estimation with Adversarial Training

Generally, it is desirable to have a model with a smooth conditional output distribution with respect to its input because most measured phenomena are inherently smooth. This idea of distributional smoothing has been used as a regularization technique by encouraging a model to be less overconfident, for example, using label smoothing [67,68] or virtual adversarial training [69].

For uncertainty estimation, distributional smoothing can improve the quality of the predictive uncertainty depending on the direction of smoothing. For example, smoothing along a random direction can be less effective while being computationally expensive in all directions. Lakshminarayanan et al. [5] propose using adversarial training with the fast gradient sign method [70] to smooth the predictive distribution along the direction where the loss is high. Qin et al. [71] investigate the relationship between adversarial robustness and predictive uncertainty. They show that inputs that are sensitive to adversarial perturbations are more likely to have unreliable predictive uncertainty. Based on this insight, they propose a new training approach that smooths training labels based on their input adversarial robustness.

In this paper, we instead propose using the “free” adversarial method [72], which recycles the gradient information from regular training to quickly generate adversarial data. Thus, we locally smooth the prediction distribution along the adversarial direction with virtually no additional cost. Figure 23 illustrates the improvement in uncertainty estimation when using adversarial training in PM-value regression task, while Figure 24 in threshold exceedance classification (using MC dropout as an example). Generally, we observe that adversarial training improves the NLL and CE (i.e., making less overconfident predictions) with negligible effects on other metrics. This, of course, depends on the size of the adversarial perturbation, which can be tuned accordingly. We observe that increasing the perturbation size can have adverse effects on the accuracy metrics, which is expected [73]. We also observe that adversarial training led to more improvements in the PM-value regression than in the threshold exceedance classification. This is because threshold exceedance events are rare, and smoothing the predictive distribution does not help catch these events.

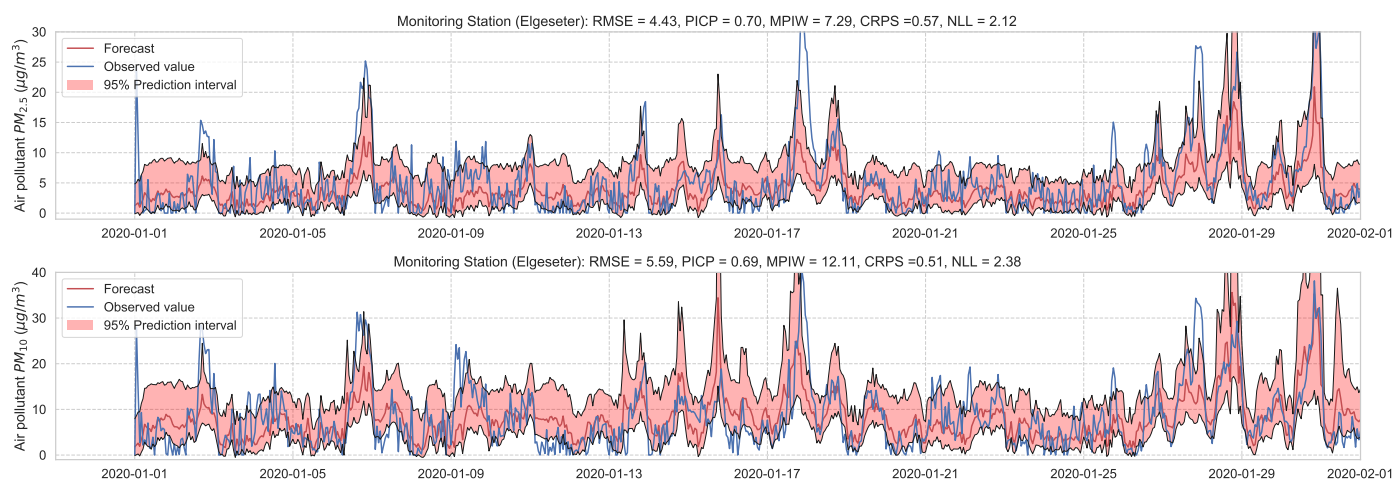


Figure 23. Cont.

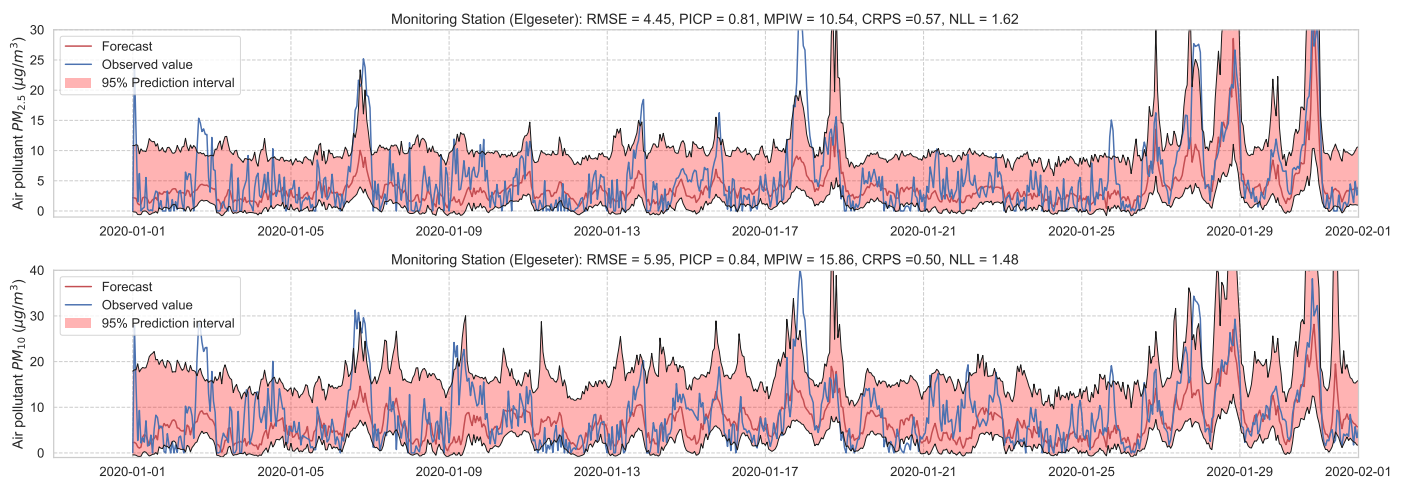


Figure 23. Comparison of uncertainty estimation in PM-value regression when training (**top**) without adversarial training versus (**bottom**) without adversarial training. Using Adversarial training leads to smoother predictive distribution; thus, lower NLL (less overconfident predictions).

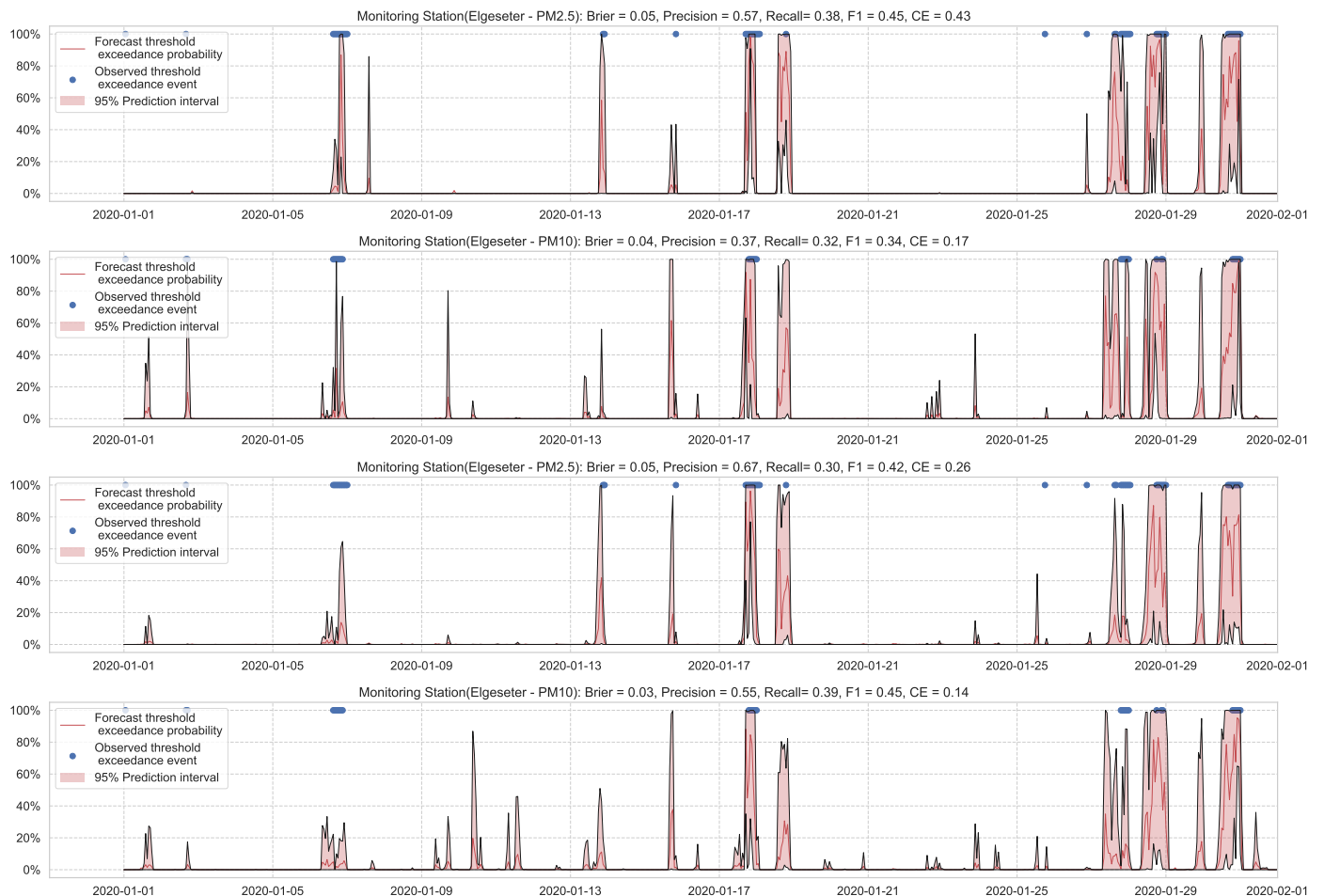


Figure 24. Comparison of uncertainty estimation in threshold exceedance classification when training (**top**) without adversarial training versus (**bottom**) without adversarial training. Using Adversarial training leads to smoother predictive distribution; thus, lower CE (less overconfident predictions).

5. Discussion

In this section, we investigate some of the implications of the study. In particular, we perform a comparative analysis to evaluate the selected probabilistic models based on empirical performance, reliability of confidence estimate, and practical applicability. Then we close by investigating the practical impact of uncertainty quantification on decision-making.

5.1. Empirical Performance

In Section 4, we summarized the performance of each model in a tabular format. Here, we compare all the models according to their empirical performance in a single monitoring station.

Figure 25 shows a comparative summary of empirical performance in the PM-value regression task. We observe that all models perform consistently well with slight variations. The BNN model performs better in metrics that assess the quality of a probabilistic forecast (i.e., in CRPS and NLL). This is expected since BNNs provide the closest approximation to Bayesian inference, while other models provide only a crude approximation. Interestingly, the GNNs with MC dropout perform very closely to BNNs in CRPS and NLL. By scoring better in CRPS, BNNs also score better in accuracy metrics (i.e., RMSE) since the CRPS generalizes the MAE to a probabilistic setting.

The PICP and MPIW are conflicting metrics that simultaneously assess the quality of the generated prediction interval. For example, by increasing the width of a prediction interval (higher MPIW), more values will be inside the predicted intervals (higher PICP). Thus, we observe that BNNs perform well in PICP but poorly in MPIW.

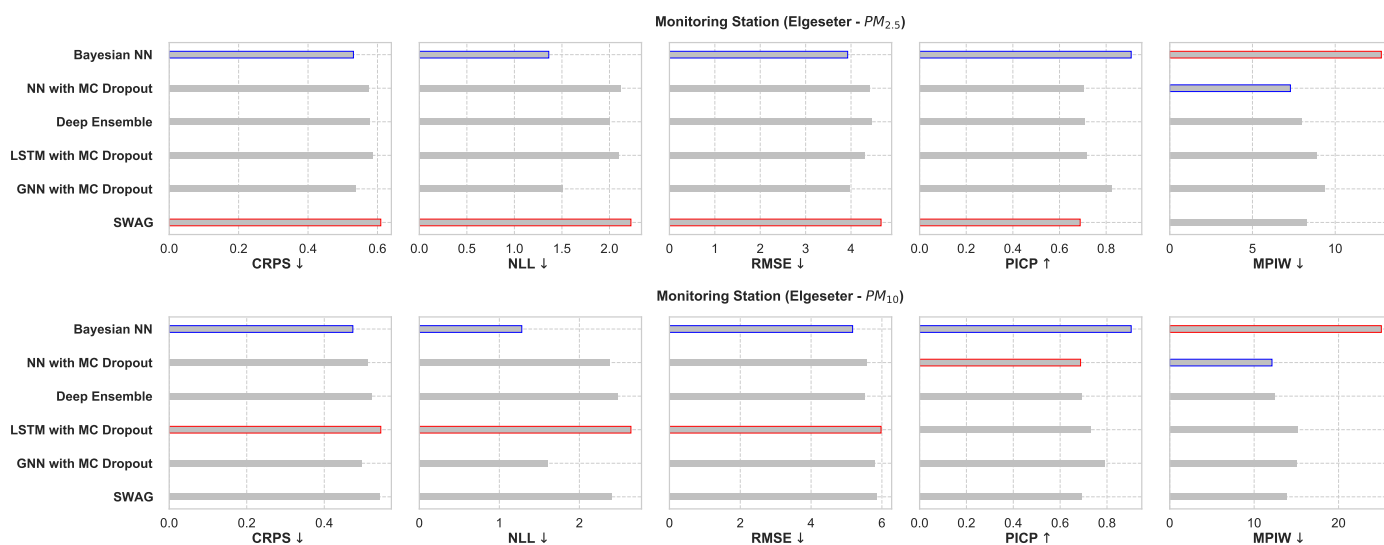


Figure 25. Comparison of empirical performance of the selected probabilistic models in the PM-value regression task. The comparison is according to five performance metrics (left to right): CRPS, NLL, RMSE, PICP, and MPIW. Blue highlights the best performance, while red highlights the worst performance. The arrows alongside the metrics indicate which direction is better for that specific metric.

Figure 26 shows a comparative summary of empirical performance in the threshold exceedance classification task. We observe that the BNN model performs better in metrics that measure the quality of probabilistic predictions (scoring rule): Brier score and cross-entropy. We also observe that the performance is inconclusive in metrics intended for deterministic classification (F1, precision, recall). This shows that these metrics are not appropriate for probabilistic prediction. Additionally, these metrics are biased by class imbalance since threshold exceedance is a rare event.

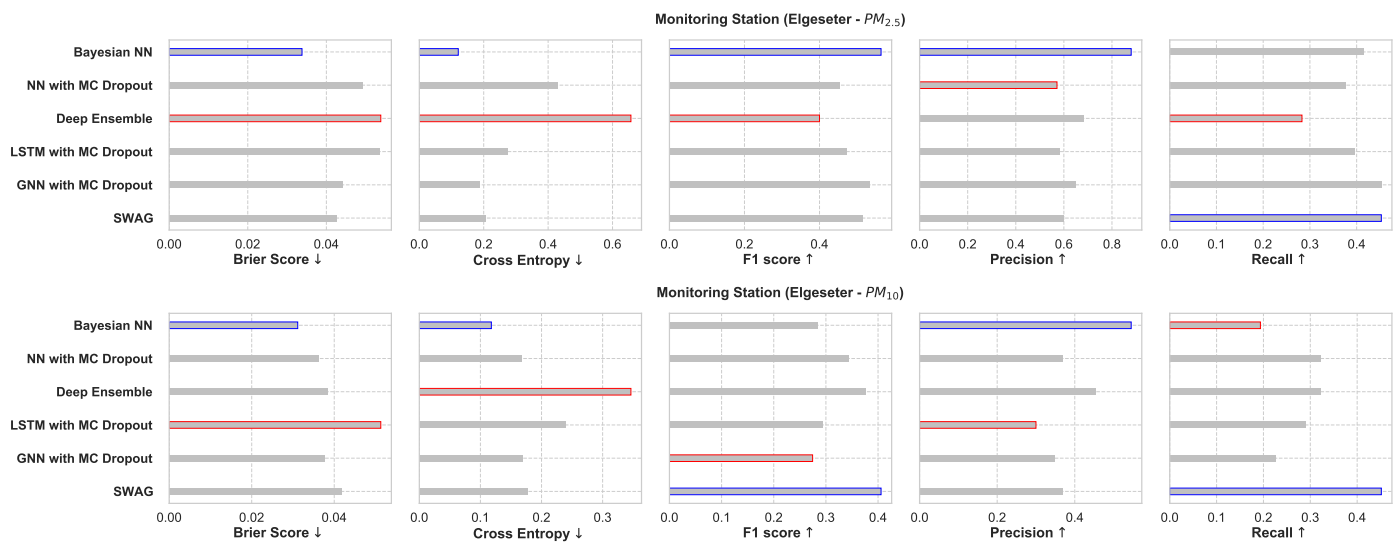


Figure 26. Comparison of empirical performance of the selected probabilistic models in the threshold exceedance classification task. The comparison is according to five performance metrics (left to right): Brier score, cross-entropy, FI score, precision, and recall. Blue highlights the best performance, while red highlights the worst performance.

5.2. Reliability of Confidence Estimate

In Section 5.1, we evaluate the selected models using metrics of accuracy and predictive probabilities separately. However, for decision-making, it is crucial to avoid overconfident, incorrect predictions. Therefore, evaluating the reliability of a confidence estimate is indispensable when selecting probabilistic models. One approach to evaluate reliability is to measure the amount of loss (or the number of incorrect predictions) a model makes when its confidence is above a certain threshold. This is a slight variation of the *accuracy-versus-confidence* technique suggested by Lakshminarayanan et al. [5] for the classification task.

For the threshold classification task, the confidence interval is bounded between 0 and 1 ($0 \leq CI \leq 1$). Thus, we can define a model confidence to be: $confidence = 1 - CI$ and a confidence threshold $0 \leq \tau \leq 1$. Then we plot the number of incorrect predictions the model makes when its confidence is above τ . We expect the curve to be monotonically decreasing for a reliable confidence estimate since a rational model has fewer incorrect predictions at high confidence. Additionally, we plot the total number of predictions (in our case, number of hours in a month of test set) as a function of τ . This curve is monotonically decreasing, but the decreasing rate indicates the amount of confidence in a model. The decreasing rate would be high in a model with lower confidence since it makes fewer predictions with high confidence. Figure 27 shows plots of loss vs. confidence and count vs. confidence in the threshold classification task. We observe that the BNN model is more reliable by making fewer incorrect predictions at high confidence but that it has the lowest confidence. Appendix B shows the results in all monitoring stations.

For the PM-value regression task, the confidence interval is unbounded ($0 < CI < \infty$). However, to compare models, we can normalize and calculate their relative confidence to be bounded between 0 and 1. Then we plot the regression loss as a function of the confidence threshold τ . Figure 28 show the plots of loss vs. confidence and count vs. confidence in the PM-value regression task. We observe that the selected models produce rational behaviors and that the BNN model is more reliable but has the lowest confidence. The curves are smooth since the loss is continuous in the regression task.

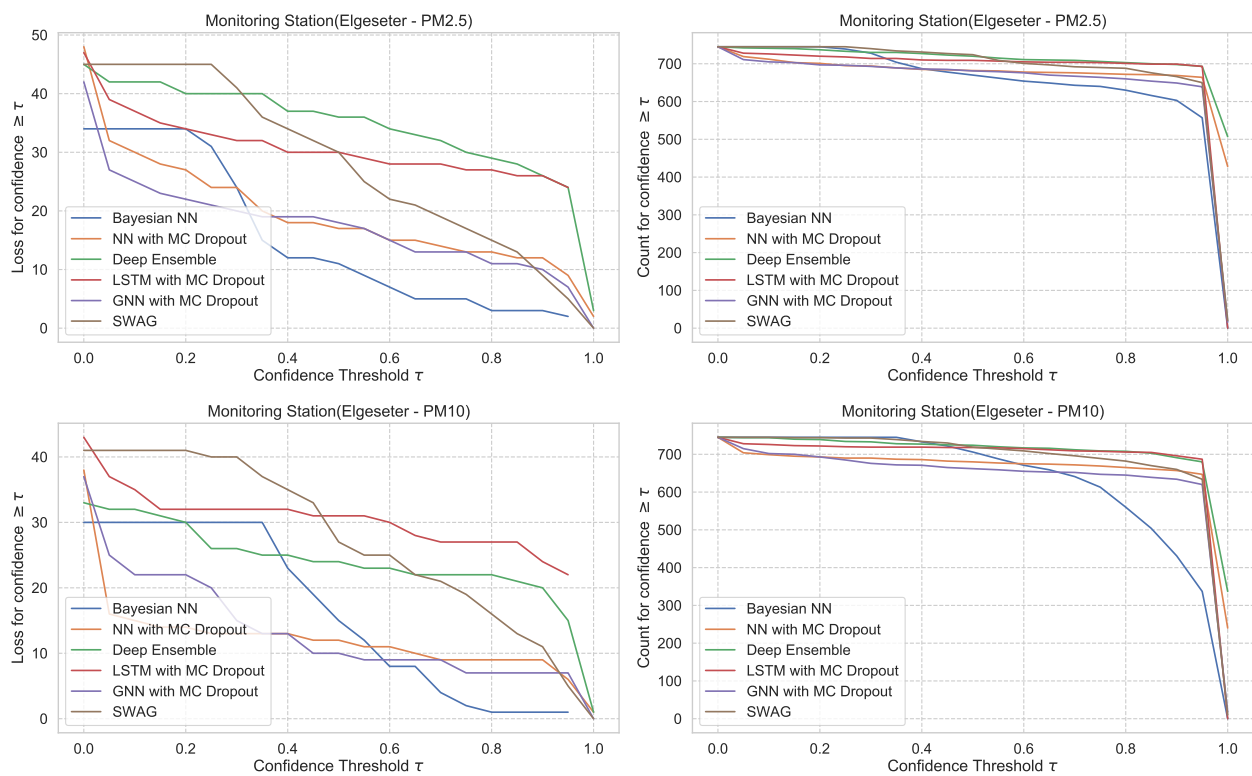


Figure 27. Comparison of confidence reliability for the selected probabilistic models in the threshold exceedance task. (Left): loss versus confidence. (Right): count versus confidence. The selected models produce are rational, which means the loss-vs-confidence curves are monotonically decreasing.

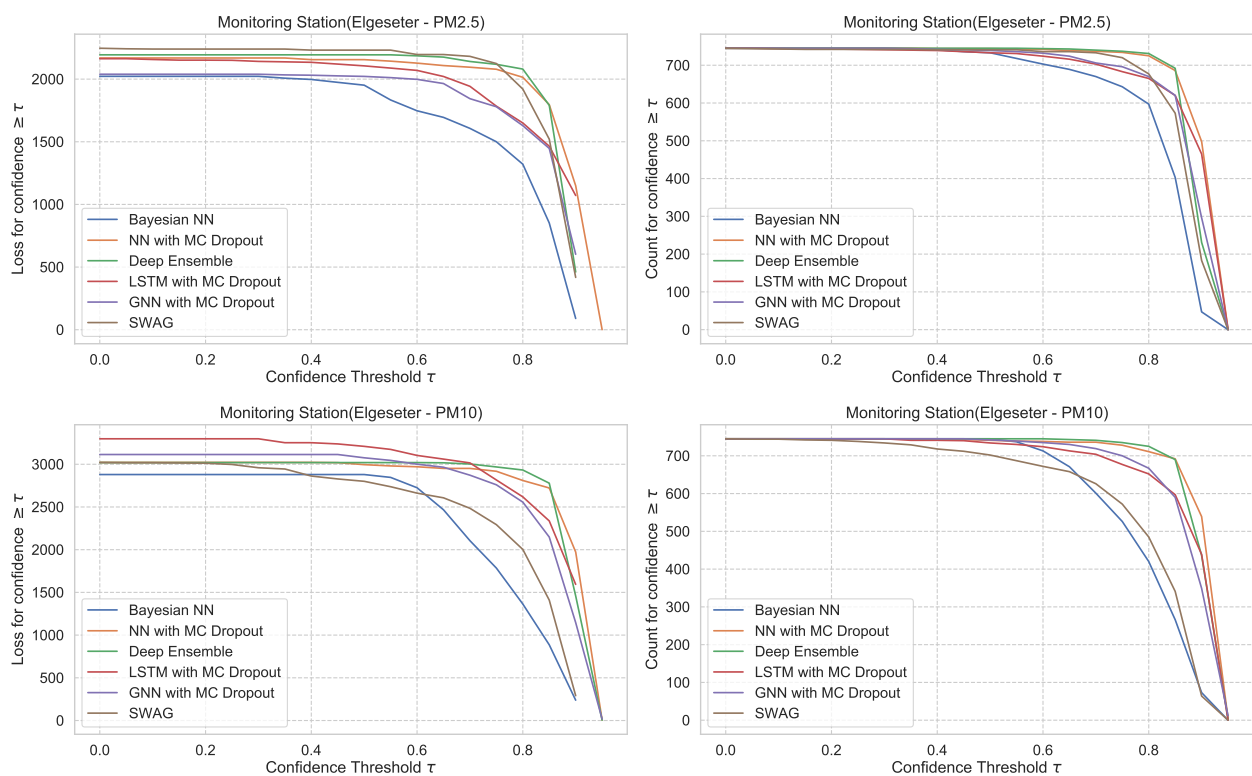


Figure 28. Comparison of confidence reliability for the selected probabilistic models in the PM-value regression task. (Left): loss versus confidence. (Right): count versus confidence.

Figure 29 shows the impact of adversarial training in the PM-value regression task, with deep ensembles as an example. We observe that adversarial training can reduce overconfident predictions.

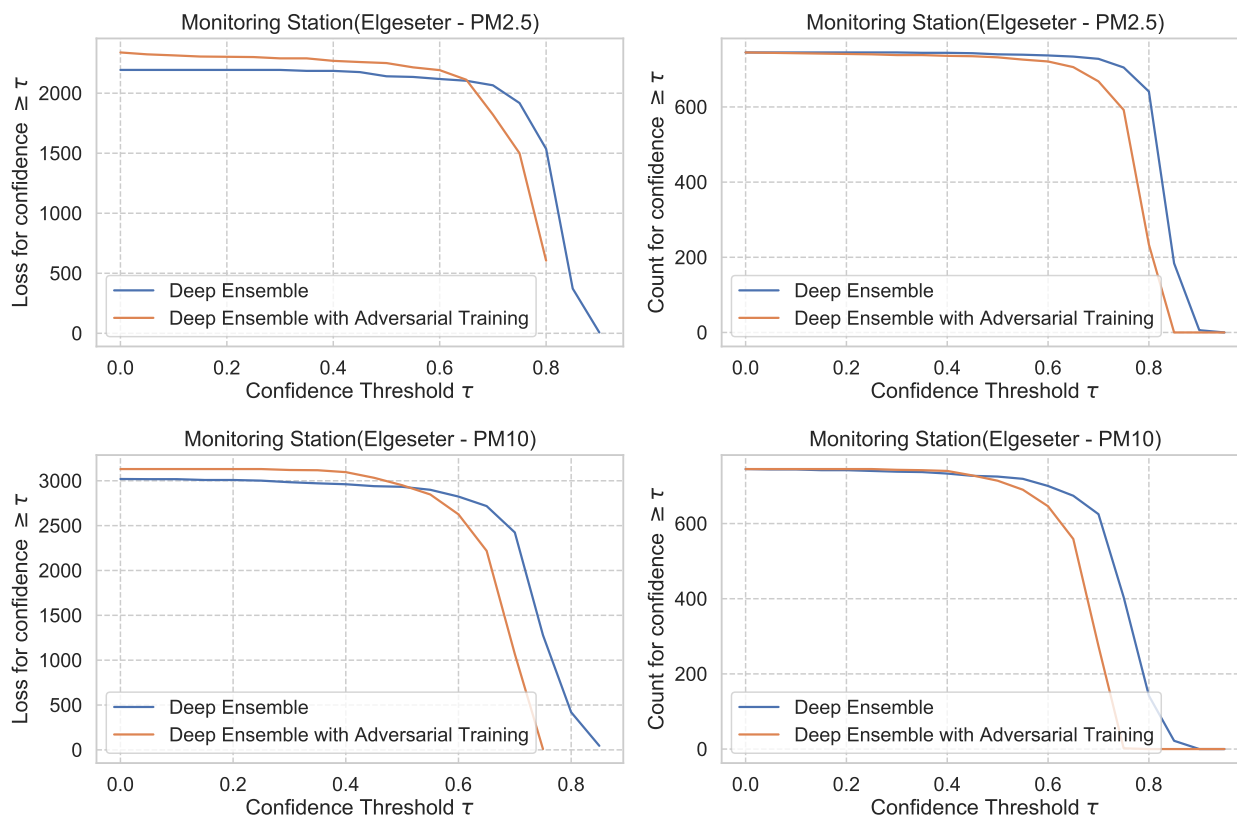


Figure 29. Impact of adversarial training on predictive uncertainty in PM-value regression, using deep ensemble as an example. **(Left:)** loss versus confidence. **(Right:)** count versus confidence.

5.3. Risk-informed Decisions

The primary motivation of quantifying epistemic uncertainty is to represent how much the model does not know. Evaluating specific decision policies is out of the scope of this paper, as they depend on the specific costs of countermeasures or the cost of high pollution values. Nonetheless, to show the value of probabilistic models, we investigate the practical impact on decision-making using a case of urban management. Recently, the Norwegian government has proposed strict regulations for particle dust thresholds. That means if the particle dust exceeds a specific threshold, the government will impose a penalty on the municipality for violating pollution regulations. Therefore, the municipality has to decide whether to implement countermeasures in advance if a forecasting model predicts threshold exceedance events.

The challenge is that not all forecasts are correct and occasionally result in false positives and false negatives. A false positive is associated with implementing unnecessary countermeasures, while a false negative is associated with violating pollution regulations. The costs of these two scenarios are context-dependent. Therefore, it is helpful to have a forecasting model that also provides potential flexibility and tradeoffs depending on the cost of false positives and false negatives.

Non-probabilistic models already offer some aspect of this tradeoff by leveraging the aleatoric uncertainty. For example, we can adjust the probability threshold in a binary classification, i.e., implement countermeasures, only if the predictive probability is above τ_1 . Figure 30a shows the decision score (in terms of F1, precision, and recall) as a function of aleatoric confidence in a non-probabilistic model. Accepting the model decision even when its predictive probability is low will result in more false positives and thus higher costs of unnecessary countermeasures. On the other hand, accepting the model decisions only when its predictive probability is high will result in more false negatives, thus increasing the risk of violating pollution regulations.

In probabilistic models, we can also leverage epistemic uncertainty to obtain a higher degree of tradeoffs. That means we implement countermeasures only if the predictive probability is above τ_1 and the model confidence is above a certain threshold τ_2 . Figure 30b shows the resulting decision score as a function of aleatoric and epistemic confidence in a probabilistic model. We use a BNN model as a representative example. To allow a fair comparison, we use a non-probabilistic model with the same architecture and trained with the same conditions. We observe that the probabilistic model provides a wider area of control over the risk profile based on the costs of false positives versus false negatives. In this case, the probabilistic model scores better over a wider range of τ_1 and τ_2 than a non-probabilistic model. This confirms that probabilistic models are more suitable for making more informed and risk-aware decisions.

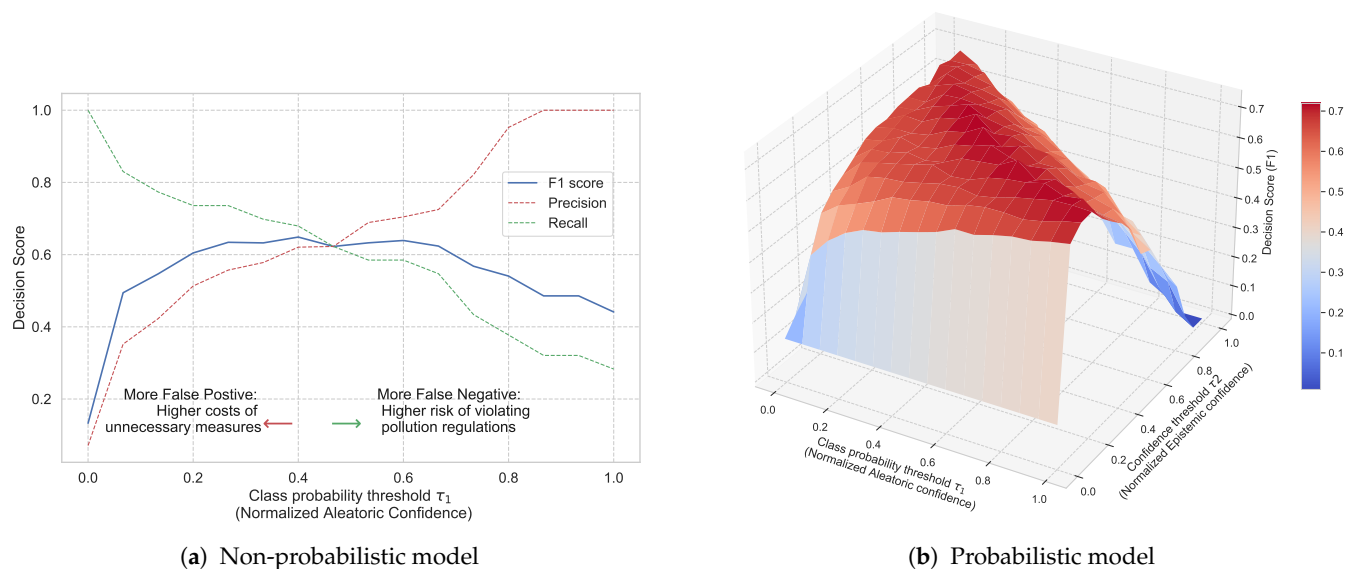


Figure 30. Comparison of decision score in non-probabilistic and probabilistic models. (a) Decision score in a non-probabilistic model as a function of class probability threshold (τ_1 corresponding to aleatoric confidence). (b) Decision score in a probabilistic model as a function of both class probability threshold (τ_1) and model confidence threshold (τ_2 corresponding to epistemic uncertainty).

5.4. Practical Applicability

In Section 5.1, we observe that a BNN model performs better in a variety of metrics. Further, BNNs offer an elegant approach to represent model uncertainty without strong assumptions. However, they are computationally demanding, require more parameters, are sensitive to hyperparameter choices, and take longer to converge than non-Bayesian neural networks. Thus, BNNs are difficult to work with, especially on larger datasets. Additionally, the nature of variational approximation and the choice of the prior or posterior distributions bias the predictive uncertainty.

From a practical perspective, MC Dropout, SWAG, and deep ensemble are convenient and scalable approaches to obtain uncertainty directly from standard neural networks without changing the optimization. Additionally, they have fewer hyperparameters and can be applied independently from the underlying network architecture (such as standard neural network, LSTM, and GNNs). However, these approaches require strong assumptions and provide a crude approximation of the Bayesian inference. During inference, they have high computational costs during inference (proportional to the number of samples M). Further, the uncertainty is estimated by measuring the diversity of predictions that depends on the network's architecture, size, and properties of the training data and initialization parameters. Accordingly, there is no guarantee that they produce a diverse prediction for an uncertain input.

The deep ensemble is a convenient non-Bayesian approach that captures aspects of multi-modality. It is better than the bootstrap ensemble in which independent models are trained using independent datasets. The deep ensemble trains every model using the whole dataset since stochastic optimization and random initialization make the models sufficiently independent. However, training deep ensemble is more expensive than SWAG or MC dropout since we need to train M different models.

The underlying assumption used in MC dropout is that it approximates a probabilistic deep Gaussian process. This assumption is based on the idea that a single-layer neural network tends to converge to a Gaussian process in the limit of an infinite number of hidden units [74]. Our results show that MC dropout provides less reliable uncertainty estimates (Figures 27 and 28), which confirms the results recently presented in other domains, such as computer vision [75] and molecular property prediction [76].

The related works that address quantifying uncertainty in a data-driven forecast of air quality uses either quantile regression (QR) [29], Gaussian processes (GP) [28], or ConvLSTM with MC dropout [26]. We have reimplemented these methods in our problem setting to compare them to the proposed models since we have different datasets and different data inputs. We use a Gradient Tree Boosting [47] for quantile regression and GPyTorch [77] for Gaussian processes. Table 8 summarizes the comparison results in the PM-value regression task in one representative monitoring station. We observe that the proposed models perform better, especially in metrics that assess the quality of a probabilistic forecast (i.e., in CRPS and NLL). Quantile regression only estimates the conditional quantile and not a probabilistic distribution. Thus, we can only assess the quality of its prediction interval and not a probabilistic forecast. Gaussian processes offer a Bayesian formalism to reason about uncertainty. However, they have the highest computational cost since they require an inversion of the kernel matrix, which has an asymptotic complexity of $\mathcal{O}(n^3)$, where n is the number of training examples.

Table 8. Comparison of the previous works and the proposed models when quantifying uncertainty in data-driven forecast of air quality

Metric	QR [29]	GP [28]	ConvLSTM [26]	BNN	Deep Ensembles	GNN	SWAG
RMSE↓	6.17	6.45	6.46	5.17	5.59	5.80	5.76
PICP↑	0.72	0.92	0.71	0.90	0.69	0.79	0.76
MPIW↓	14.79	36.96	12.18	25.07	12.11	15.07	16.61
CRPS↓	NA	0.53	0.56	0.47	0.51	0.50	0.53
NLL↓	NA	1.36	2.48	1.28	2.38	1.60	1.96

6. Conclusions

This work presents a broad empirical evaluation of the relevant state-of-the-art deep probabilistic models applied in air quality forecasting. Through extensive experiments, we describe training these models and evaluating their predictive uncertainties using various metrics for regression and classification tasks. We introduce a new state-of-the-art example for air quality forecasting by defining the problem setup and selecting proper input features and models. Then, we apply uncertainty-aware models that exploit the temporal and spatial correlation inherent in air quality data using recurrent and graph neural networks. We propose improving uncertainty estimation using "free" adversarial training to locally smooth the prediction distribution along the adversarial direction with virtually no additional cost. Finally, we show how data-driven probabilistic models can improve the current practice using a real-world example of air quality forecasting in Norway. Particularly, we show the practical impact of uncertainty quantification and demonstrate that probabilistic models are more suitable for making informed decisions.

The results show that the proposed models perform better than previous works in quantifying uncertainty in data-driven air quality forecasts. BNNs provide a more reliable uncertainty estimate but with challenging practical applicability. Additionally, the results demonstrate that MC Dropout, SWAG, and deep ensemble have less reliable uncertainty estimates, but they are more convenient in practical applicability. Our contribution hereof is not about specific model selection but more about navigating the larger design space and the different tradeoffs offered by various probabilistic models.

Future work includes exploring hybrid air quality forecasting combining physics-based and data-driven probabilistic models. Additionally, future work should address uncertainty estimation with out-of-domain or seasonal variations in air quality forecasting. In our experiments, we used a reasonable search space for hyperparameters and network architectures. Thus, future work could target an exhaustive search of hyperparameters, additional data sets, and models (e.g., transformers).

Author Contributions: A.M.: Conceptualization, investigation, methodology, software, experimentation, visualization, validation, writing—original draft. F.A.K., K.B., and G.T.: Conceptualization, investigation, methodology, writing—review and editing, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the European Union's Horizon 2020 research and innovation program, project AI4EU, grant agreement No. 825619.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset of this study is available at https://github.com/Abdulmajid-Murad/deep_probabilistic_forecast (accessed on 27 November 2021). We use the open database of air quality measurements offered by the Norwegian Institute for Air Research (NILU) (<https://www.nilu.com/open-data/>, accessed on 27 November 2021). The meteorological data are based on historical weather and climate data offered by the Norwegian Meteorological Institute (<https://frost.met.no>, accessed on 27 November 2021). The traffic data are based on aggregated traffic volumes offered by the Norwegian Public Roads Administration (<https://www.vegvesen.no/trafikkdata/start/om-api>, accessed on 27 November 2021).

Acknowledgments: We would like to thank Sigmund Akselsen for reading a draft of the paper and providing detailed feedback on the work. We also thank Tiago Veiga for helping with a script to fetch the datasets.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MC	Monte Carlo
SWAG	Stochastic weight averaging-Gaussian
IoT	Internet of Things
MACC	Monitoring Atmospheric Composition and Climate
uEME	urban European Monitoring and Evaluation Program
NB-IoT	Narrowband-IoT
LSTM	Long short-term memory
CNN	Convolutional neural network
GRU	Gated recurrent unit
GP	Gaussian process
NO ₂	Nitrogen dioxide
MCMC	Markov chain Monte Carlo
PM ₁₀	Coarse particulate matter of diameter less than 10 µm
PM _{2.5}	Fine particulate matter of diameter less than 2.5 µm
CAQI	Common air quality index
NILU	Norwegian institute for air research
CI	Confidence interval
PI	Prediction interval
XGBoost	eXtreme Gradient Boosting
QR	Quantile regression
RMSE	Root-mean-square error
CE	Cross entropy
BS	Brier Score
PICP	Prediction interval coverage probability
MPIW	Mean prediction interval width
NLL	Negative log-likelihood
CRPS	Continuous ranked probability score
CDF	Cumulative distribution function
MAE	Mean absolute error
BNN	Bayesian Neural Network
KL	Kullback–Leibler divergence
GNN	Graph neural network
GLU	Gated linear units
GCN	Graph convolutional network

Appendix A. Datasets

This section shows additional figures explaining the used dataset of air quality, meteorological data, traffic, and street-cleaning reports from the municipality.

Appendix A.1. Air Quality Data

Figure A1 shows the air quality data of PM_{2.5} and PM₁₀, measured over two years in four different sensing stations in the city of Trondheim (Figure 2). The figure shows a clear temporal and spatial correlations between the four sensing stations.

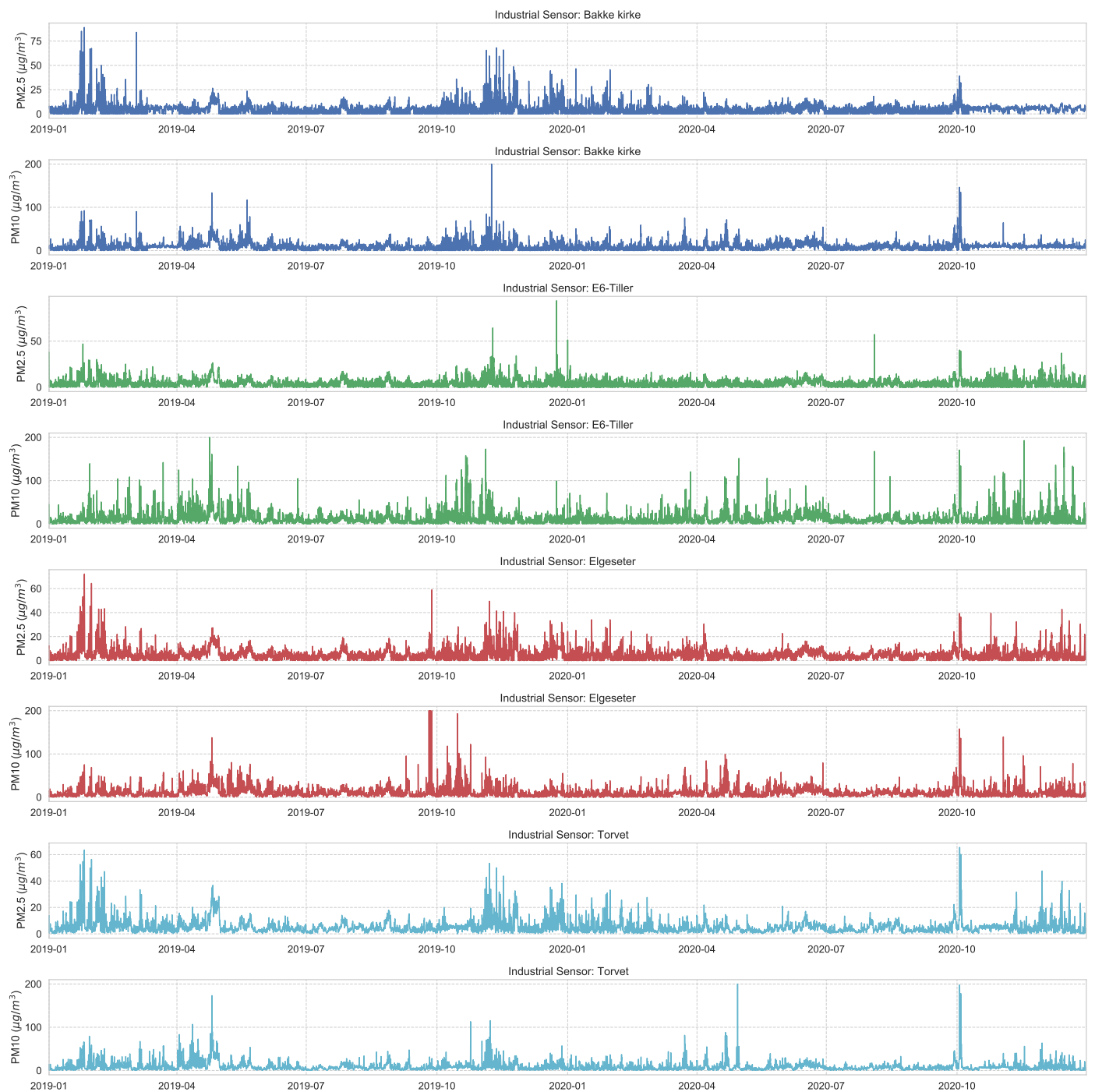


Figure A1. Air quality data of $PM_{2.5}$ and PM_{10} , measured over two years in four different sensing stations in the city of Trondheim. These data are offered by the Norwegian Institute for Air Research (NILU) (<https://www.nilu.com/open-data/>, accessed on 27 November 2021).

Appendix A.2. Weather Data

We use weather data observations at four monitoring stations in the city of Trondheim (Voll, Sverreborg, Gloshaugen, Lade). These observations include: air temperature, relative humidity, precipitation, air pressure, wind speed, wind direction, snow thickness, and duration of sunshine. Figure A2 summarize all these observations.

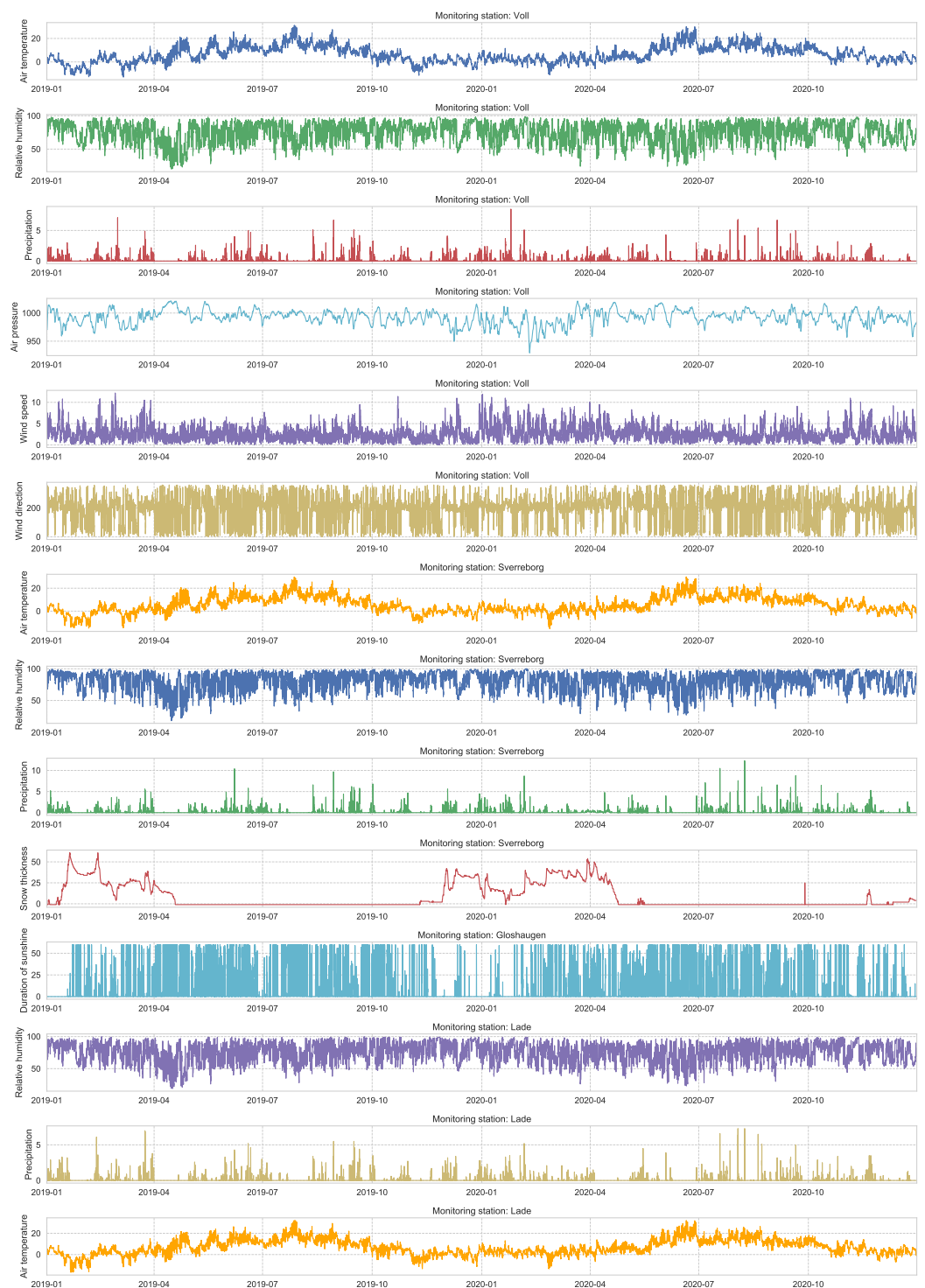


Figure A2. Weather data observations over two years at four monitoring station in the city of Trondheim (Voll, Sverreborg, Gloschaugen, Lade). These data are offered by the Norwegian Meteorological Institute (<https://frost.met.no>, accessed on 27 November 2021).

Appendix A.3. Traffic Data

We also used traffic volume as an input feature for the forecasting models. We used the data of the traffic volume recorded at eight streets of Trondheim. Figure A3 shows the traffic volume recorded over two years. The figure shows a clear correlation of traffic between streets. It also shows the drop in traffic during the pandemic lockdown in Trondheim (after 12 March 2020).

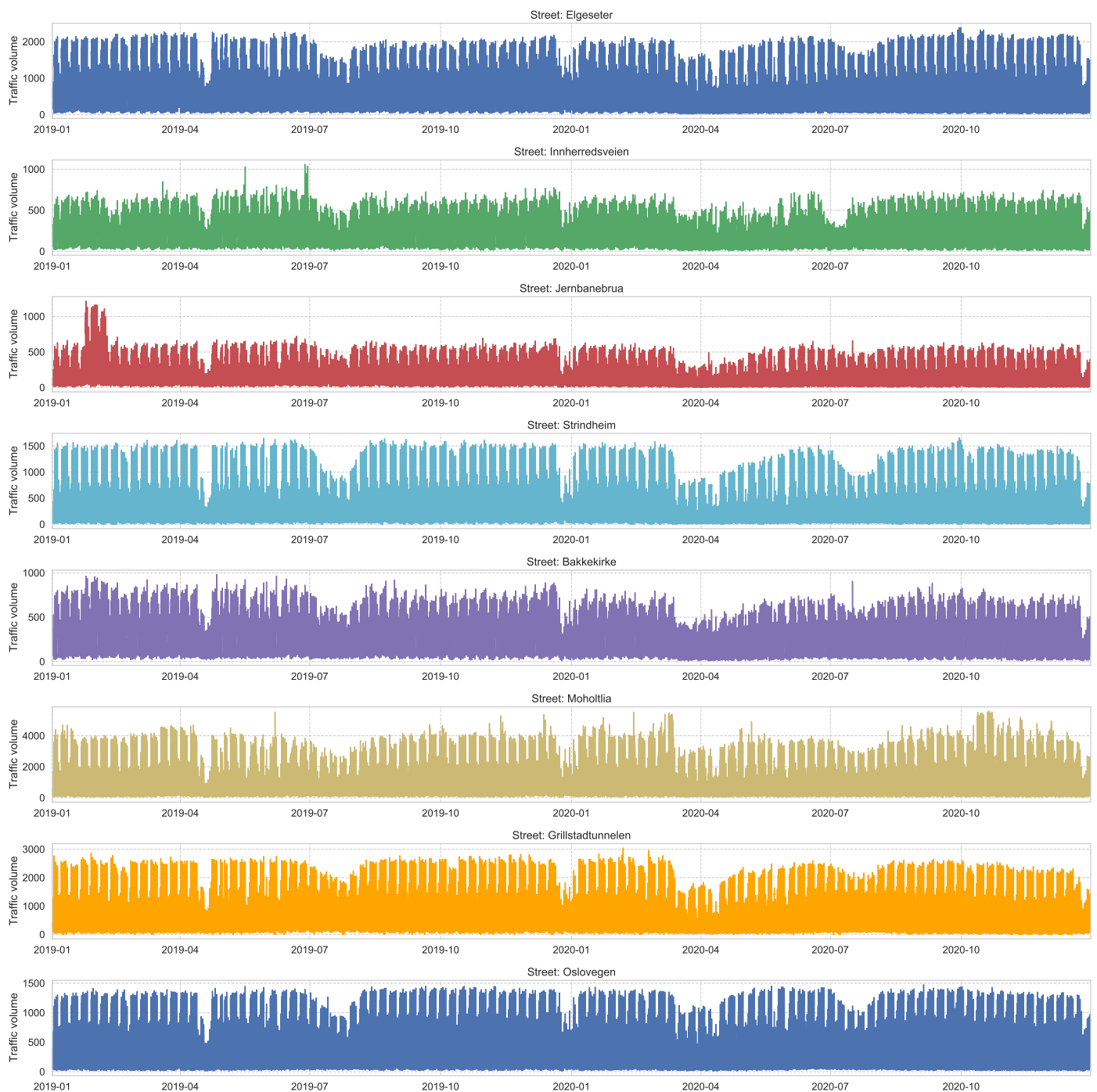


Figure A3. Traffic volume recorded at eight streets of Trondheim over two years. These data are offered by the Norwegian Public Roads Administration (<https://www.vegvesen.no/trafikdata/start/om-api>, accessed on 27 November 2021).

We also used data of street-cleaning at main streets of Trondheim, as shown in Figure A4. These data are reported by the municipality and include the duration of time in which a street-cleaning is taking place.

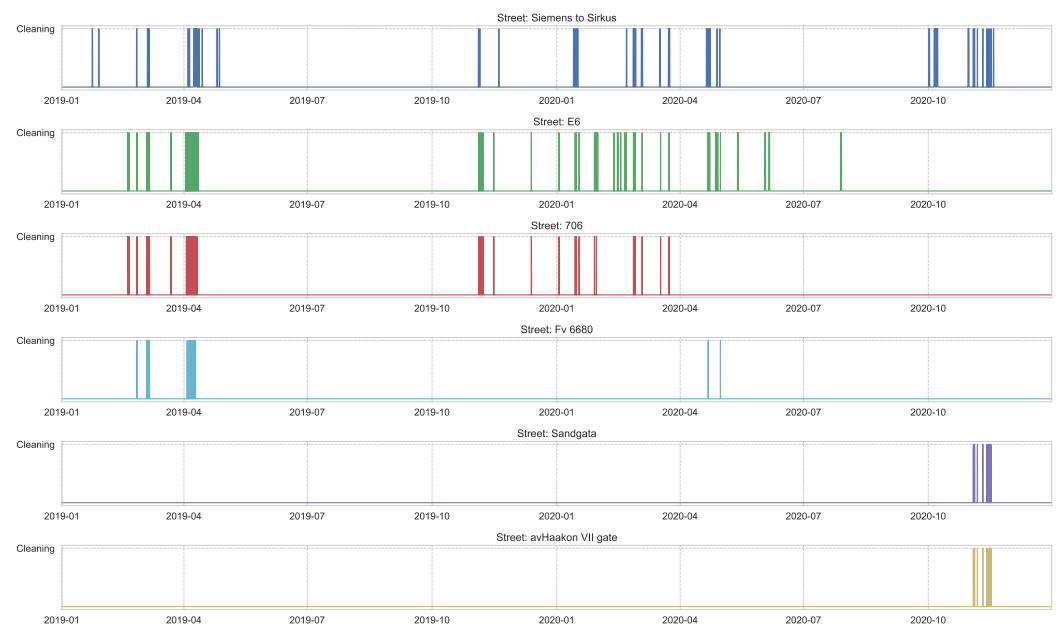


Figure A4. Data of the duration of time in which a street-cleaning is taking place on the main streets of Trondheim, reported by the municipality.

Appendix B. Reliability of Confidence Estimate: Additional Plots

This section presents additional plots for comparison of confidence reliability in all monitoring stations. Figure A5 shows the comparison of confidence reliability in the PM-value regression task while Figure A6 in the threshold exceedance classification task.

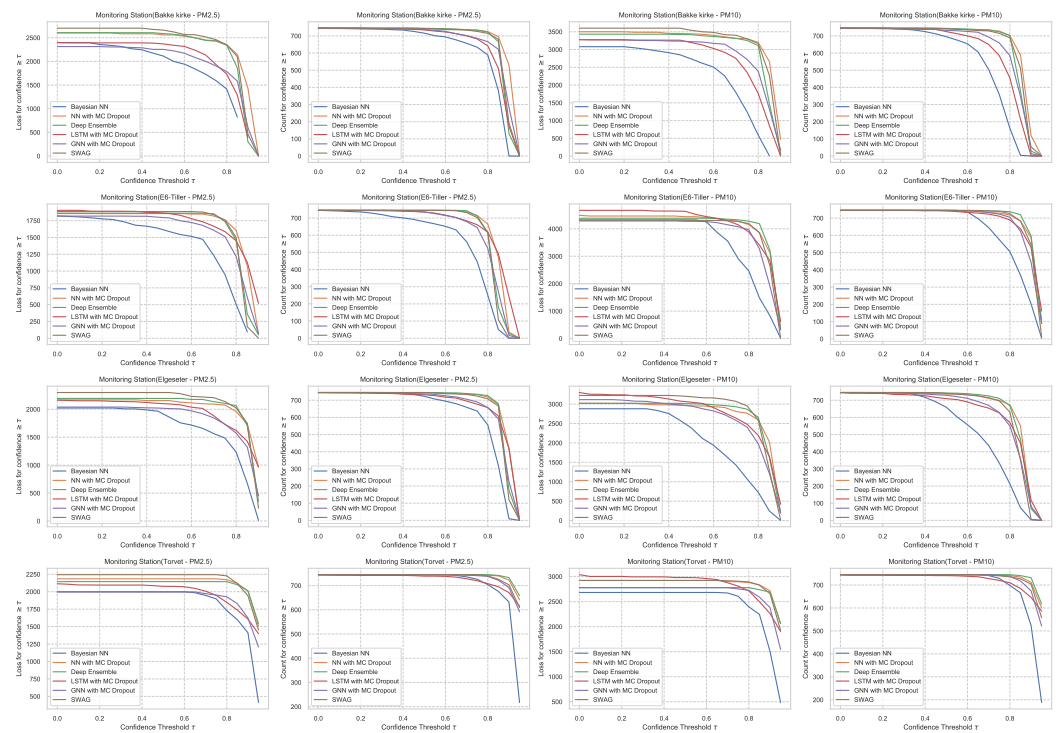


Figure A5. Comparison of confidence reliability for the selected probabilistic models in the PM-value regression task in all monitoring stations.

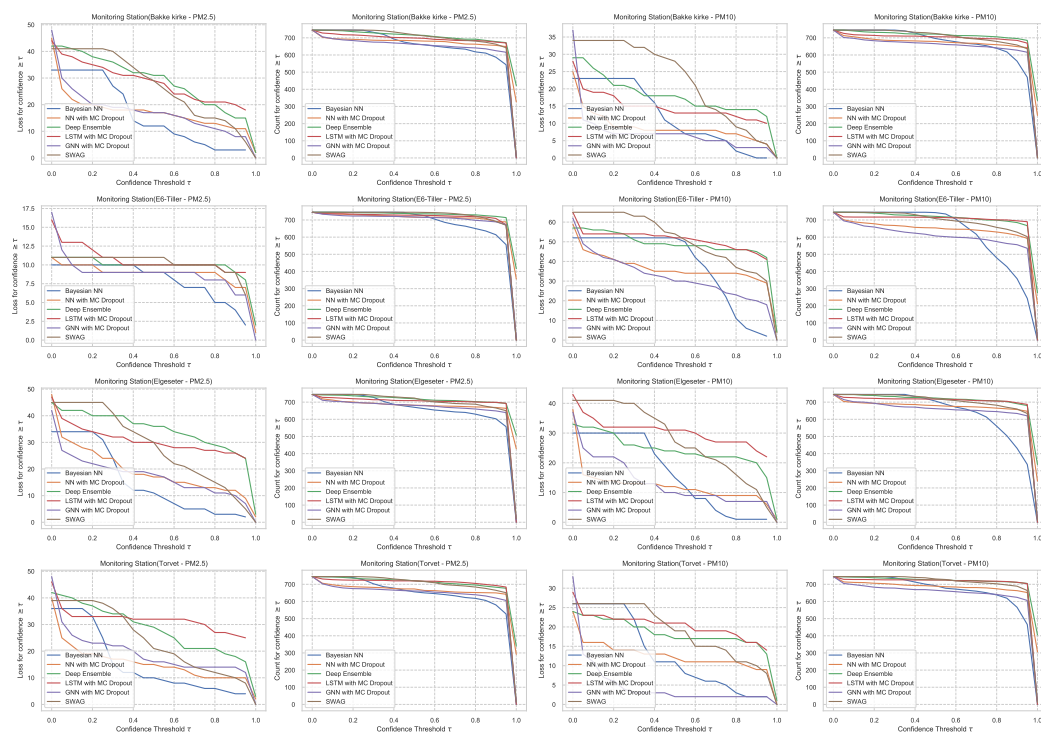


Figure A6. Comparison of confidence reliability for the selected probabilistic models in the threshold exceedance classification task in all monitoring stations.

Appendix C. Justification for Threshold-Exceedance Classification

Recently, the Norwegian government has proposed even stricter regulations for particle dust thresholds. Therefore, we want to estimate if the concentration of air particles exceeds certain thresholds following the Common Air Quality Index (CAQI) used in Europe [36], as shown in Table 1.

The CAQI specifies five levels of air pollutants, but the air quality in the city of Trondheim is usually at *Very Low* and rarely exceeds the *Medium* level. Figure A8 shows the air quality level over one year in all monitoring stations. Generally, air quality data have right-skewed distributions, with the degree of skewness (asymmetry) differing among different cities. Figure A7 shows histograms that approximately represent the distribution of air quality at four monitoring stations in the city of Trondheim. The figure clearly shows heavily right-skewed distributions, in which higher CAQI classes are under-represented. Therefore, instead of a multinomial classification task (with five classes), we transform the problem into a threshold exceedance forecast task in which we try to predict the points in time where the air quality exceeds the *Very Low* level.

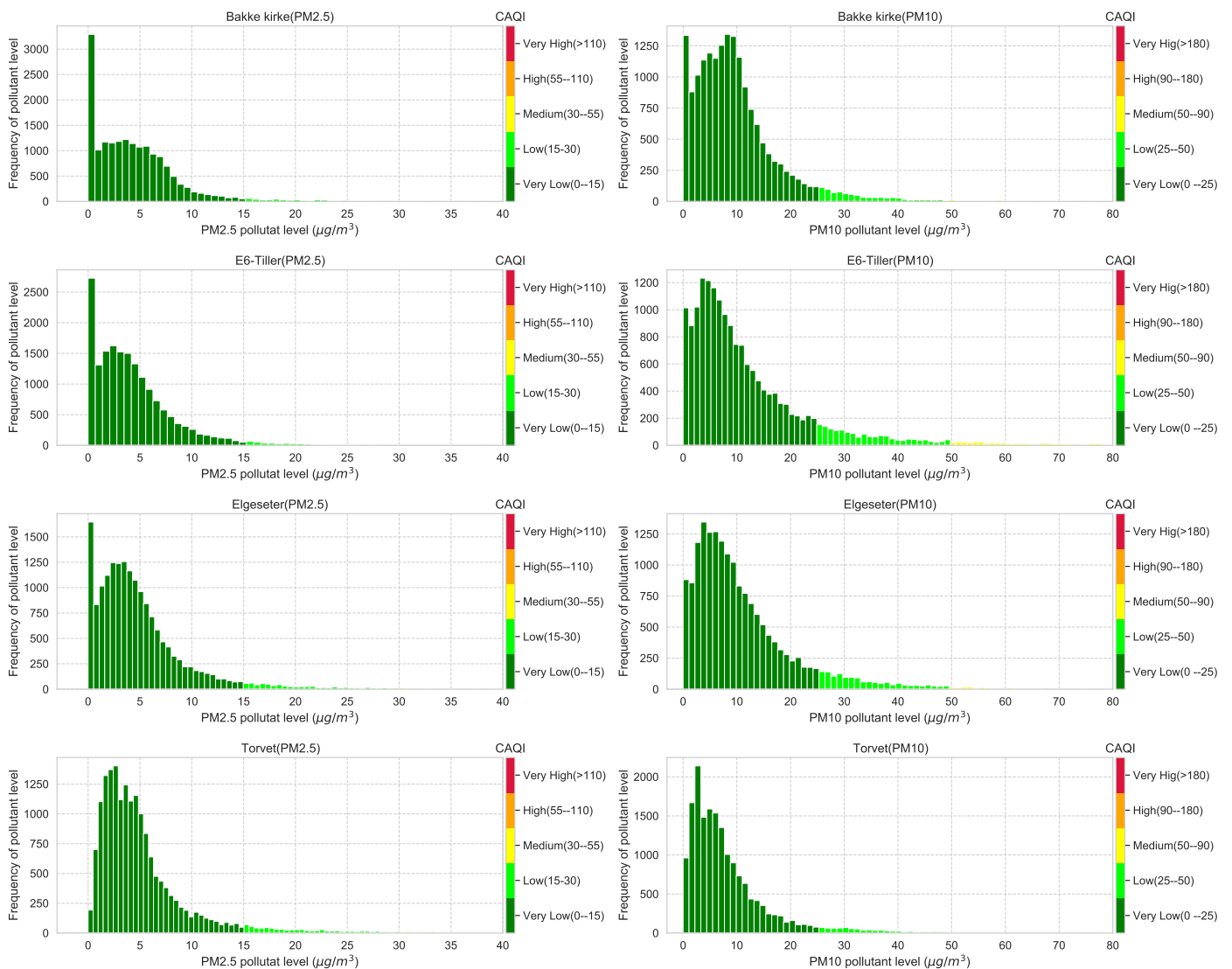


Figure A7. Histograms that approximately represent the distribution of air quality at four monitoring stations in the city of Trondheim. The air quality data come from heavily right-skewed distributions, in which higher CAQI classes are under-represented.

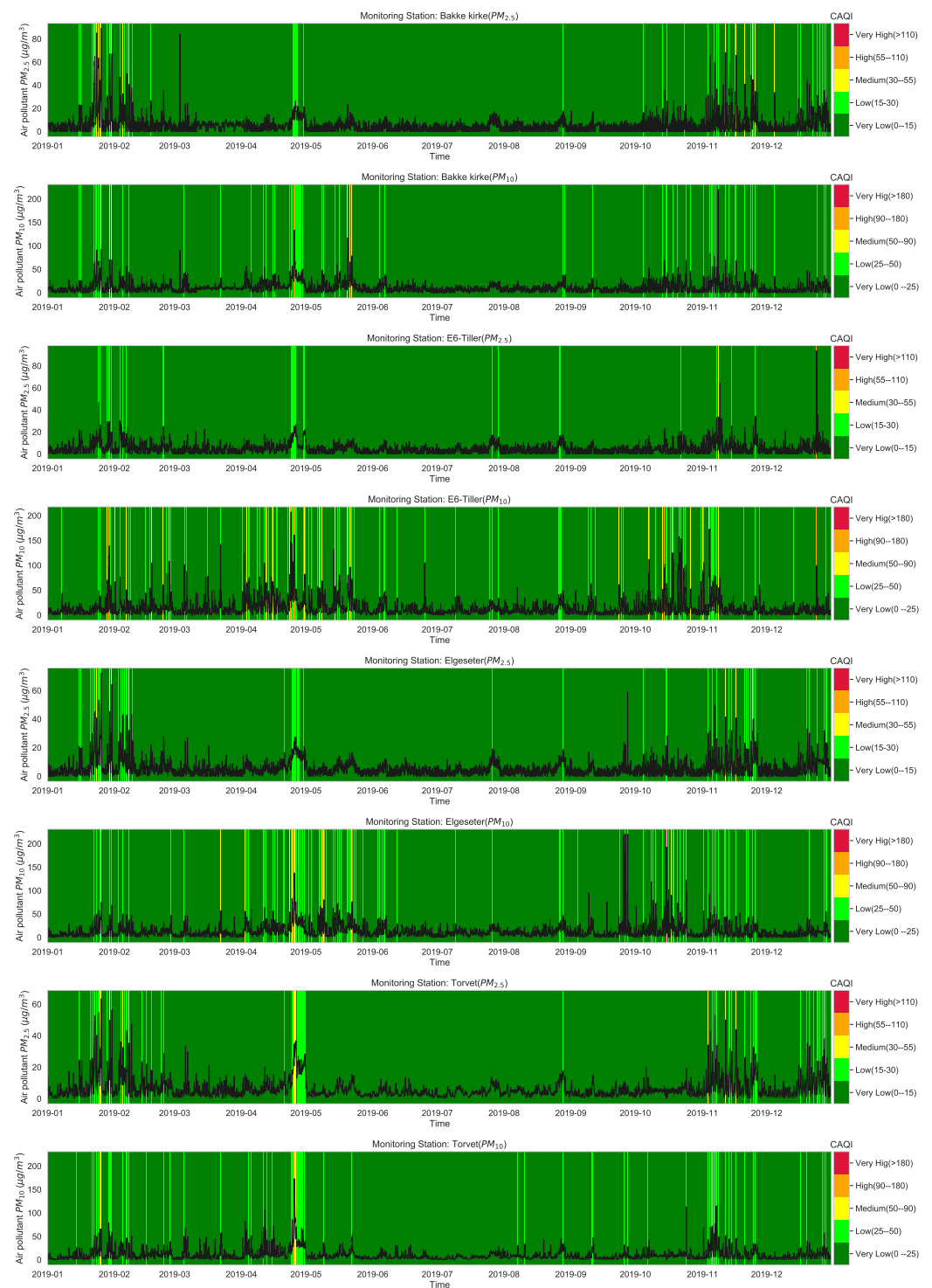


Figure A8. Air quality in the city of Trondheim over one year in all monitoring stations. The data are decomposed into the five CAQI levels of air pollutants. It is usually at *Very Low* and rarely exceeds the *Medium* level.

Appendix D. Experimental Details of Deep Probabilistic Forecasting Models

In this section, we show the results of the PM-value regression and threshold exceedance classification in all monitoring stations using the corresponding probabilistic model.

Appendix D.1. Bayesian Neural Networks (BNNs)

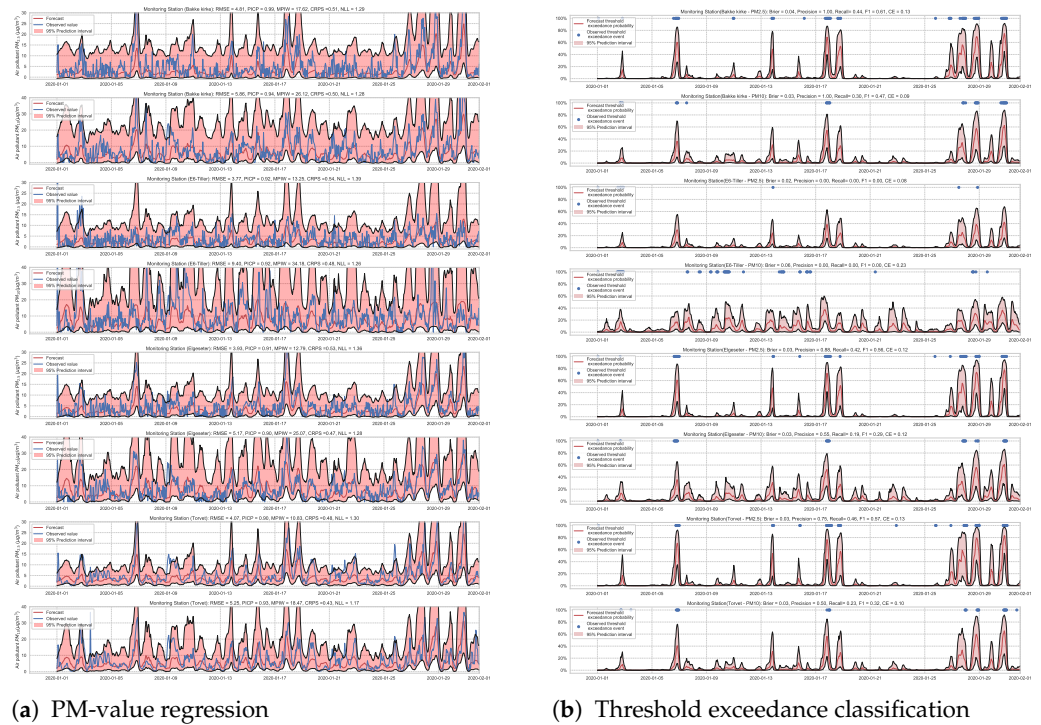


Figure A9. Deep probabilistic forecast of air quality using BNNs at four monitoring stations.

Appendix D.2. Standard NNs with MC Dropout

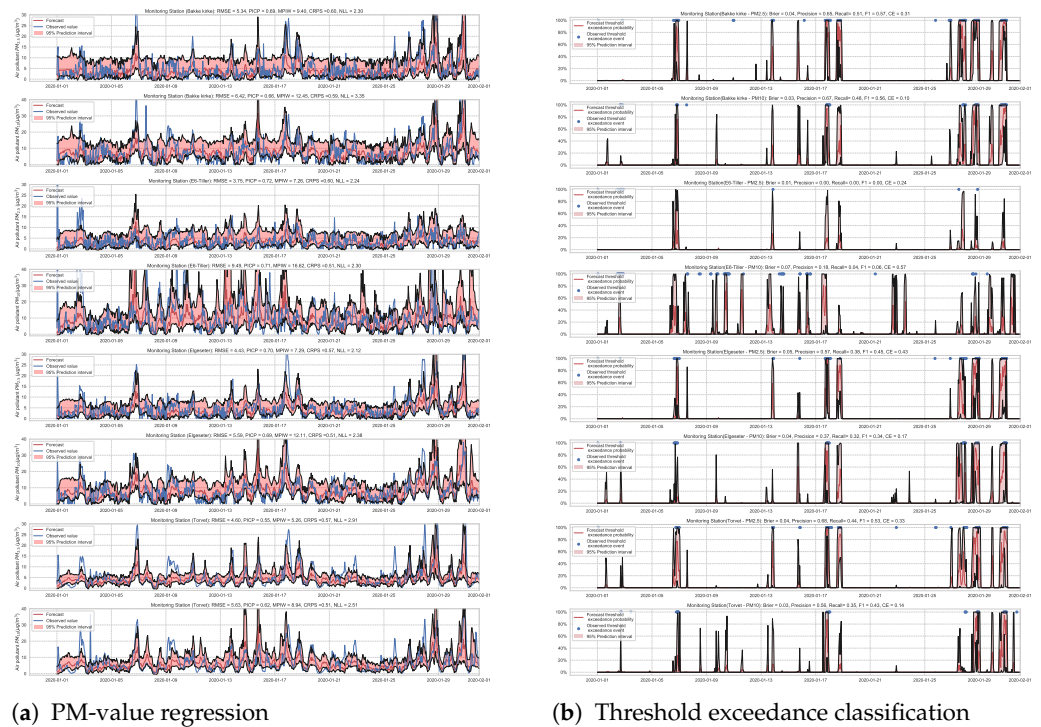


Figure A10. Deep probabilistic forecast of air quality using NNs with MC dropout at four monitoring stations.

Appendix D.3. Deep Ensemble

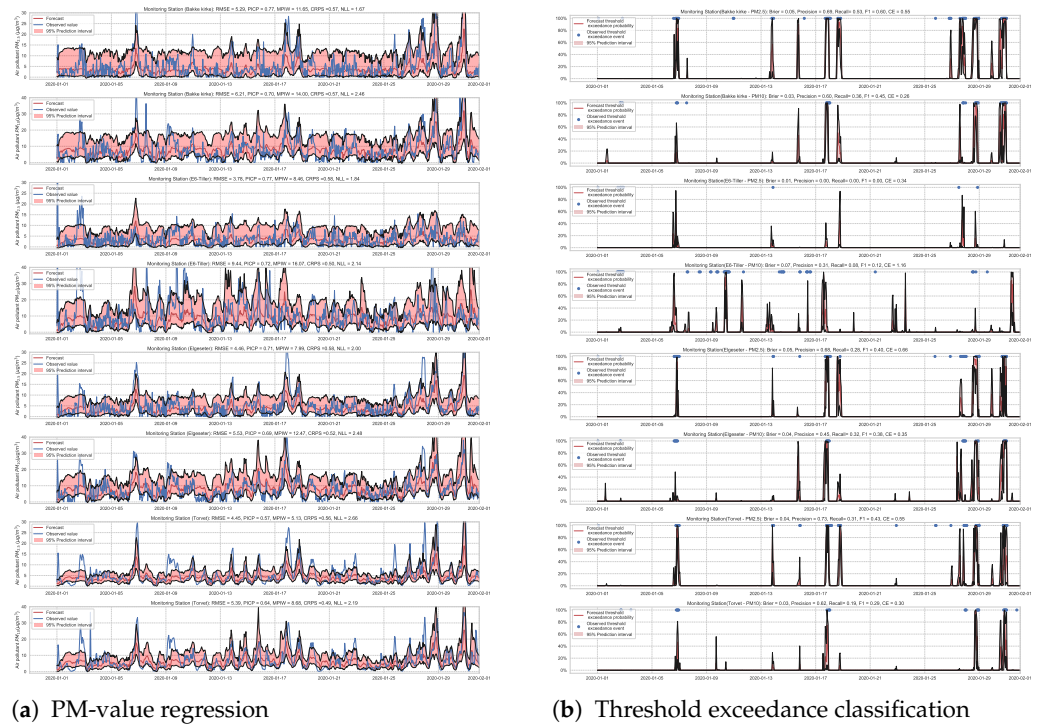


Figure A11. Deep probabilistic forecast of air quality using deep ensemble at four monitoring stations.

Appendix D.4. LSTM with MC Dropout

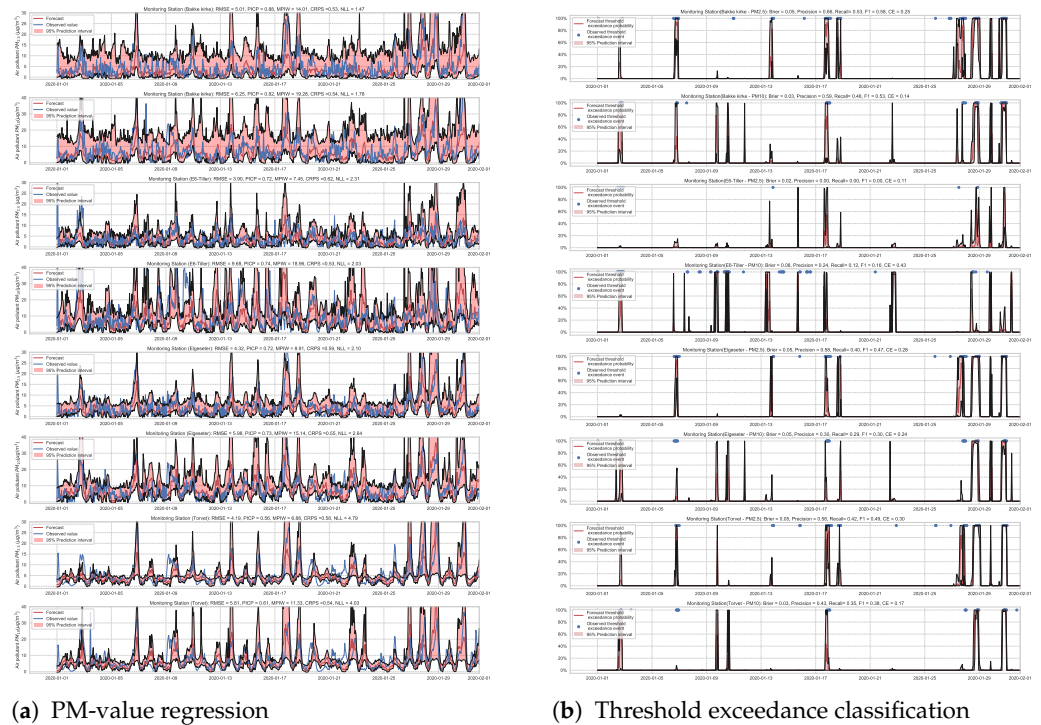


Figure A12. Deep probabilistic forecast of air quality using LSTM with MC dropout at four monitoring stations.

Appendix D.5. GNNs with MC Dropout

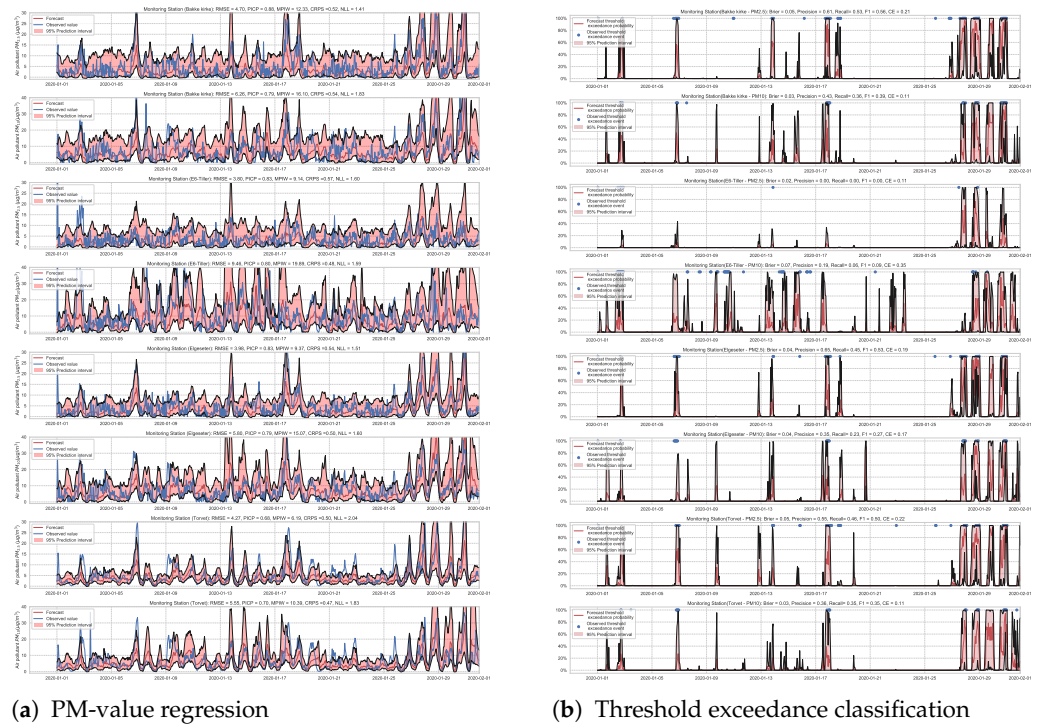


Figure A13. Deep probabilistic forecast of air quality using GNNs with MC dropout at four monitoring stations.

Appendix D.6. SWAG

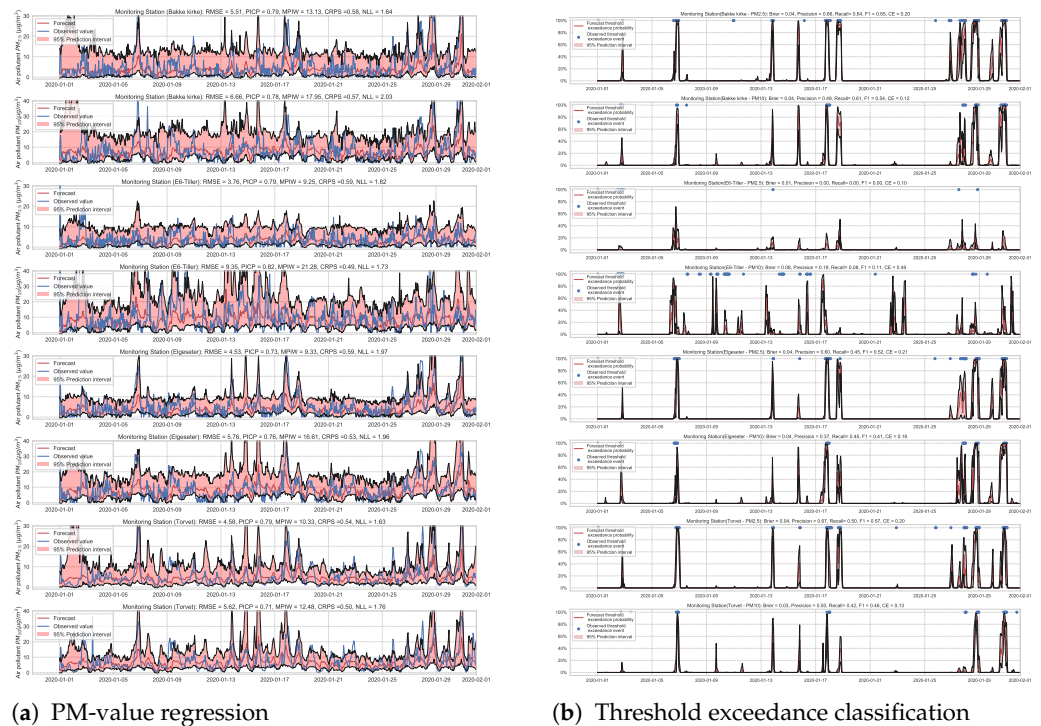


Figure A14. Deep probabilistic forecast of air quality using a SWAG model at four monitoring stations.

References

1. Ghahramani, Z. Probabilistic machine learning and artificial intelligence. *Nature* **2015**, *521*, 452–459. [CrossRef] [PubMed]
2. MacKay, D.J. A practical bayesian framework for backpropagation networks. *Neural Comput.* **1992**, *4*, 448–472. [CrossRef]
3. Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Wierstra, D. Weight uncertainty in neural network. In proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 1613–1622.
4. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1050–1059.
5. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Proceedings of the International Conference on Neural Information Processing Systems, California, CA, USA, 4–9 December 2017; pp. 6405–6416.
6. Zhu, L.; Laptev, N. Deep and Confident Prediction for Time Series at Uber. In proceedings of 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 103–110.
7. Maddox, W.J.; Izmailov, P.; Garipov, T.; Vetrov, D.P.; Wilson, A.G. A simple baseline for bayesian uncertainty in deep learning. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December, 2019; pp. 13153–13164.
8. Kendall, A.; Gal, Y.; Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceeding of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Utah, UT, USA, 18–23 June 2018; pp. 7482–7491.
9. Kendall, A.; Gal, Y. What uncertainties do we need in Bayesian deep learning for computer vision? In Proceedings of the International Conference on Neural Information Processing Systems, California, CA, USA, 4–9 December 2017; pp. 5580–5590.
10. Chien, J.T.; Ku, Y.C. Bayesian recurrent neural network for language modeling. *IEEE Trans. Neural Networks Learn. Syst.* **2016**, *27*, 361–374. [CrossRef] [PubMed]
11. Xiao, Y.; Wang, W.Y. Quantifying uncertainties in natural language processing tasks. In proceedings of the AAAI Conference on Artificial Intelligence, Hawaii, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7322–7329.
12. Ott, M.; Auli, M.; Grangier, D.; Ranzato, M. Analyzing uncertainty in neural machine translation. In proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 3956–3965.
13. Meyer, G.P.; Thakurdesai, N. Learning an uncertainty-aware object detector for autonomous driving. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 10521–10527.
14. Marécal, V.; Peuch, V.H.; Andersson, C.; Andersson, S.; Arteta, J.; Beekmann, M.; Benedictow, A.; Bergström, R.; Bessagnet, B.; Cansado, A.; et al. A regional air quality forecasting system over Europe: the MACC-II daily ensemble production. *Geosci. Model Dev.* **2015**, *8*, 2777–2813. [CrossRef]
15. Walker, S.E.; Hermansen, G.H.; Hjort, N.L. Model selection and verification for ensemble based probabilistic forecasting of air pollution in Oslo, Norway. In Proceedings of the 60th ISI World Statistics Congress (WSC), Rio de Janeiro, Brazil, 26–31 July 2015.
16. Garaud, D.; Mallet, V. Automatic calibration of an ensemble for uncertainty estimation and probabilistic forecast: Application to air quality. *J. Geophys. Res. Atmos.* **2011**, *116*. [CrossRef]
17. Air Quality Forecasting Service in Norway. Available online: <https://luftkvalitet.miljodirektoratet.no/kart/59/10/5/aqi> (accessed on 27 November 2021).
18. Denby, B.R.; Gauss, M.; Wind, P.; Mu, Q.; Grøtting Wærsted, E.; Fagerli, H.; Valdebenito, A.; Klein, H. Description of the uEMEP_v5 downscaling approach for the EMEP MSC-W chemistry transport model. *Geosci. Model Dev.* **2020**, *13*, 6303–6323. [CrossRef]
19. Mu, Q.; Denby, B.R.; Wærsted, E.G.; Fagerli, H. Downscaling of air pollutants in Europe using uEMEP_v6. *Geosci. Model Dev. Discuss.* **2021**, 1–24. [CrossRef]
20. Norman, M.; Sundvor, I.; Denby, B.R.; Johansson, C.; Gustafsson, M.; Blomqvist, G.; Janhäll, S. Modelling road dust emission abatement measures using the NORTRIP model: Vehicle speed and studded tyre reduction. *Atmos. Environ.* **2016**, *134*, 96–108. [CrossRef]
21. Denby, B.R.; Klein, H.; Wind, P.; Gauss, M.; Pommier, M.; Fagerli, H.; Valdebenito, A. The Norwegian Air Quality Service: Model Forecasting. Available online: https://wiki.met.no/_media/airquip/workshopno/denby_17sep2018.pdf (accessed on 27 November 2021).
22. Simpson, D.; Benedictow, A.; Berge, H.; Bergström, R.; Emberson, L.D.; Fagerli, H.; Flechard, C.R.; Hayman, G.D.; Gauss, M.; Jonson, J.E.; et al. The EMEP MSC-W chemical transport model—technical description. *Atmos. Chem. Phys.* **2012**, *12*, 7825–7865. [CrossRef]
23. Lepperød, A.; Nguyen, H.T.; Akselsen, S.; Wienhofen, L.; Øzturk, P.; Zhang, W. Air Quality Monitor and Forecast in Norway Using NB-IoT and Machine Learning. *Int. Summit Smart City 360°*. Springer: New York, NY, USA, 2019; pp. 56–67.
24. Veiga, T.; Munch-Ellingsen, A.; Papastergiopoulos, C.; Tzovaras, D.; Kalamaras, I.; Bach, K.; Votis, K.; Akselsen, S. From a Low-Cost Air Quality Sensor Network to Decision Support Services: Steps towards Data Calibration and Service Development. *Sensors* **2021**, *21*, 3190. [CrossRef]
25. Zhou, Y.; Chang, F.J.; Chang, L.C.; Kao, I.F.; Wang, Y.S. Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts. *J. Clean. Prod.* **2019**, *209*, 134–145. [CrossRef]

26. Mokhtari, I.; Bechkit, W.; Rivano, H.; Yaici, M.R. Uncertainty-Aware Deep Learning Architectures for Highly Dynamic Air Quality Prediction. *IEEE Access* **2021**, *9*, 14765–14778. [[CrossRef](#)]
27. Tao, Q.; Liu, F.; Li, Y.; Sidorov, D. Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU. *IEEE Access* **2019**, *7*, 76690–76698. [[CrossRef](#)]
28. Pucer, J.F.; Pirš, G.; Štrumbelj, E. A Bayesian approach to forecasting daily air-pollutant levels. *Knowl. Inf. Syst.* **2018**, *57*, 635–654.
29. Aznarte, J.L. Probabilistic forecasting for extreme NO2 pollution episodes. *Environ. Pollut.* **2017**, *229*, 321–328. [[CrossRef](#)]
30. Graves, A. Practical variational inference for neural networks. In proceedings of the International Conference on Neural Information Processing Systems, Granada, Spain, 12–15 December 2011; pp. 2348–2356.
31. Louizos, C.; Welling, M. Multiplicative normalizing flows for variational bayesian neural networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2218–2227.
32. Neal, R.M. *Bayesian learning for Neural Networks*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
33. Welling, M.; Teh, Y.W. Bayesian learning via stochastic gradient Langevin dynamics. In Proceedings of the International Conference on Machine Learning, Washington, WA, USA, 28 June–2 July 2011; pp. 681–688.
34. Chen, T.; Fox, E.; Guestrin, C. Stochastic gradient hamiltonian monte carlo. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1683–1691.
35. Ritter, H.; Botev, A.; Barber, D. A scalable laplace approximation for neural networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; Volume 6.
36. Elshout, S.v.d.; Léger, K. CAQI Air quality index—Comparing Urban Air Quality across Borders-2012. Technical Report, EUROPEAN UNION European Regional Development Fund Regional Initiative Project, 2012. Available online: https://www.airqualitynow.eu/download/CITEAIR-Comparing_Urban_Air_Quality_across_Borders.pdf (accessed on 27 November 2021).
37. Open Database of Air Quality Measurements by the Norwegian Institute for Air Research (NILU). Available online: <https://www.nilu.com/open-data/> (accessed on 27 November 2021).
38. The Meteorological Data by the Norwegian Meteorological Institute. Available online: <https://frost.met.no> (accessed on 27 November 2021).
39. Traffic Data by the Norwegian Public Roads Administration. Available online: <https://www.vegvesen.no/trafikkdata/start/om-api> (accessed on 27 November 2021).
40. Heskes, T.; Wiegerinck, W.; Kappen, H. Practical confidence and prediction intervals for prediction tasks. *Prog. Neural Process.* **1997**, *8*, 128–135. [[CrossRef](#)]
41. Dar, Y.; Muthukumar, V.; Baraniuk, R.G. A Farewell to the Bias-Variance Tradeoff? An Overview of the Theory of Overparameterized Machine Learning. *arXiv* **2021**, arXiv:2109.02355.
42. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, California, CA, USA, 13–17 August 2016; pp. 785–794.
43. Gneiting, T.; Raftery, A.E. Strictly proper scoring rules, prediction, and estimation. *J. Am. Stat. Assoc.* **2007**, *102*, 359–378. [[CrossRef](#)]
44. Brier, G.W. Verification of forecasts expressed in terms of probability. *Mon. Weather. Rev.* **1950**, *78*, 1–3. [[CrossRef](#)]
45. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [[CrossRef](#)]
46. Koenker, R.; Hallock, K.F. Quantile regression. *J. Econ. Perspect.* **2001**, *15*, 143–156. [[CrossRef](#)]
47. Friedman, J.H. Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
48. Nix, D.A.; Weigend, A.S. Estimating the mean and variance of the target probability distribution. In Proceedings of the IEEE International Conference on Neural Networks, Florida, FL, USA, 28 June–2 July 1994; Volume 1, pp. 55–60.
49. Hoffman, M.D.; Blei, D.M.; Wang, C.; Paisley, J. Stochastic variational inference. *J. Mach. Learn. Res.* **2013**, *14*, 1303–1347.
50. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
51. Kingma, D.P.; Salimans, T.; Welling, M. Variational dropout and the local reparameterization trick. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, Canada, 7–12 December 2015; Volume 2, pp. 2575–2583.
52. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
53. Dietterich, T.G. Ensemble methods in machine learning. In Proceedings of the International Workshop on Multiple Classifier Systems, Cagliari, Italy, 21–23 June 2000; pp. 1–15.
54. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. In Proceedings of The International Conference on Neural Information Processing Systems (Deep Learning and Representation Learning Workshop), Montreal, Canada, 8–13 December 2014.
55. Jozefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; Wu, Y. Exploring the limits of language modeling. *arXiv* **2016** arXiv:1602.02410.
56. Chen, R.; Wang, X.; Zhang, W.; Zhu, X.; Li, A.; Yang, C. A hybrid CNN-LSTM model for typhoon formation forecasting. *Geoinformatica* **2019**, *23*, 375–396. [[CrossRef](#)]
57. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.

58. Murad, A.; Pyun, J.Y. Deep recurrent neural networks for human activity recognition. *Sensors* **2017**, *17*, 2556. [[CrossRef](#)]
59. Chimmula, V.K.R.; Zhang, L. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos Solitons Fractals* **2020**, *135*, 109864. [[CrossRef](#)]
60. Sak, H.; Senior, A.W.; Beaufays, F. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling: Research.Google. Available online: <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/43905.pdf> (accessed on 27 November 2021).
61. Gal, Y.; Ghahramani, Z. A theoretically grounded application of dropout in recurrent neural networks. In Proceedings of the International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 1027–1035.
62. Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. In Proceedings of the International Conference on Neural Information Processing Systems, British Columbia, Canada, 6–12 December 2020; Volume 33, pp. 17766–17778.
63. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 933–941.
64. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
65. Hasanzadeh, A.; Hajiramezanali, E.; Boluki, S.; Zhou, M.; Duffield, N.; Narayanan, K.; Qian, X. Bayesian graph neural networks with adaptive connection sampling. In Proceedings of International Conference on Machine Learning, Virtual, Vienna, Austria, 12–18 July 2020; pp. 4094–4104.
66. Izmailov, P.; Podoprikin, D.; Garipov, T.; Vetrov, D.; Wilson, A.G. Averaging weights leads to wider optima and better generalization. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, California, CA, USA, 6–10 August 2018; pp. 876–885.
67. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
68. Muller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 4694–4703.
69. Miyato, T.; Maeda, S.i.; Koyama, M.; Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1979–1993. [[CrossRef](#)]
70. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In proceedings of the International Conference on Machine Learning, Lille, France, 7–9 July 2015.
71. Qin, Y.; Wang, X.; Beutel, A.; Chi, E.H. Improving uncertainty estimates through the relationship with adversarial robustness. *arXiv* **2020**, arXiv:2006.16375. Available online: <https://arxiv.org/abs/2006.16375> (accessed on 27 November 2021).
72. Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L.S.; Taylor, G.; Goldstein, T. Adversarial training for free! In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, Canada, 8–14 December 2019; pp. 3358–3369.
73. Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Proceedings of the International Conference on Machine Learning, California, CA, USA, 9–15 June 2019; pp. 7472–7482.
74. Williams, C.K. Computing with infinite networks. In Proceedings of the International Conference on Neural Information Processing Systems, Denver, CO, USA, 3–5 December 1996; pp. 295–301.
75. Gustafsson, F.K.; Danelljan, M.; Schon, T.B. Evaluating scalable bayesian deep learning methods for robust computer vision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 318–319.
76. Scalia, G.; Grambow, C.A.; Pernici, B.; Li, Y.P.; Green, W.H. Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *J. Chem. Inf. Model.* **2020**, *60*, 2697–2717. [[CrossRef](#)] [[PubMed](#)]
77. Gardner, J.R.; Pleiss, G.; Bindel, D.; Weinberger, K.Q.; Wilson, A.G. Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, Canada, 2–8 December 2018; pp. 7576–7586.