Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Urban Air Pollution Inference: A Neural Network Ensemble Approach

Master Thesis

Martin Lendi
lendim@student.ethz.ch

Computer Engineering and Networks Laboratory
Department of Information Technology and Electrical Engineering
ETH Zürich

Supervisors:
Balz Maag
Dr. Zimu Zhou
Prof. Dr. Lothar Thiele

August 27, 2018

Acknowledgements

I would like to thank my supervisors Dr. Zimu Zhou and Balz Maag for their great support during the master thesis. Furthermore, I would like to thank the Computer Engineering and Networks Laboratory for the provided workplace and equipment and for giving me the opportunity to work on this thesis. I would also like to thank Benno Knapp from the traffic control center of Zurich and Orhan Özkul from the office of transport of the canton Zurich for the provided traffic data. Further I thank Henrik Becker from the Institute of Transport Planning and Systems of ETH Zurich for the help and recommendations for aggregating traffic data.

Abstract

Air pollution maps can help people to plan outdoor activities and environmental scientists to evaluate new policies. Urban air is filled with ultrafine particles (UFPs) that can have a severe impact on human health. Therefore, UFP concentration maps with high spatio-temporal resolution are of great importance to control air pollution.
An extensive amount of UFP data is provided over more than three years by sensors mounted on trams of Zurich as part of the OpenSense project. However, the locations of these measurements are only sparsely distributed over the city area as the mobility of the trams is limited. Air quality information at locations without any measurements can be inferred by annotating measurement data with information about factors that have an impact on air pollution like weather, traffic and land use. In this thesis, an air pollution prediction approach based on neural networks is proposed and tested on the OpenSense dataset. Timely dynamic weather and traffic features are aggregated in order to improve the temporal resolution of urban air pollution maps. In order to check the performance of the model at locations where no UFP measurements are available, two approaches that quantify the uncertainty of air pollution predictions from neural networks based on ensembles and dropout are proposed and evaluated.
The comparison of the neural network with a generalized additive model showed that similar or better performance in terms of the error metrics is achieved. However, unrealistic pollution concentrations are predicted at locations that differ a lot in terms of environmental conditions from the training dataset. This observation is confirmed by the generated uncertainty maps.

Contents

Chapter 1

Introduction

1.1 Problem Description

Air pollution maps are getting more and more important in urban areas, as a high concentration of air pollutants can have a severe impact on human health. In particular, a high particle number concentration of ultrafine particles (UFPs), which include all particles with a diameter of less than 100 nm, can have adverse health effects. Such air pollution maps also raise people's awareness about air pollution and they empower environmental scientists to evaluate new policies. Traditionally, air quality data is measured by stationary measurement stations. However, with these static stations it is not possible to capture the high spatially and temporal variability of air pollutants. That is why several sensors were deployed on top of trams as part of the OpenSense project [1] in order to get air quality data for a wider part of Zurich. This mobile sensor network delivers a big amount of spatially and temporally high-resoluted air pollution measurements.
Urban air quality depends on multiple factors, such as land use (e.g. industrial activities, population density and height of buildings), traffic or meteorology. The annotation of the data measured along the tram lines with these factors allows pollutant concentration inference. Using this approach, the air pollution concentration can be estimated at locations without any measurements and an air pollution map can be constructed. Unfortunately, the dependencies of the explanatory variables and air pollution concentration is mostly unknown. Therefore, it is crucial to find a model that is able to capture the influence of various feature combinations effectively.
One possibility to create air pollution maps is the land-use regression (LUR) model [2]. This model is presented in the following section.

1.2 Land-use regression (LUR) model

The proposed LUR model (Figure 1.1) evaluates the dependency between a set of explanatory variables (land-use and traffic data) and the measured UFP con-

centrations to model the pollution concentration on a grid with 13200 cells (each 100m x 100m) of Zurich. Generalized additive models (GAMs) are used in order to capture the non-linear relationships between the measured concentration and the explanatory variables. For each time scale (yearly, seasonal, monthly, biweekly, weekly, daily and semi-daily), a separate model using the following relationship is built:

$$\ln(c_{num}) = a + s_1(A_1) + s_2(A_2) + \cdots + s_n(A_n) + \epsilon.$$

Here, c_{num} denotes the UFP concentration, a the intercept, $s_1 \cdots s_n$ the smoothing functions, $A_1 \cdots A_n$ the explanatory variables and ϵ the error term.
As a result, a decent performance could be reached for the models with yearly to weekly time scales. However, the accuracy of pollution maps created for daily and semi-daily temporal resolutions is notably worse compared with yearly to weekly maps. The main problem is that for temporal high-resolution maps, much less measurements are available to calculate the mean UFP concentration in a cell. This leads to a less reliable mean and a bigger impact of inaccurate measurements.

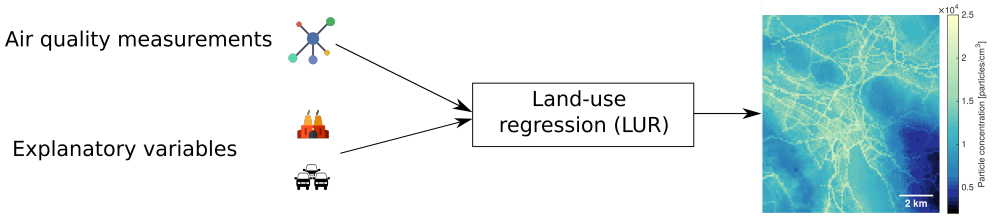


Figure 1.1: Land-use regression model

In order to increase the accuracy of highly temporally resolved pollution maps, a history database is introduced. This database includes past pollution measurements as well as its environmental conditions (e.g. temperature) and weekday. By selecting data with similar environmental conditions and weekday from this database, the number of measurements used for fitting the model is extended and the accuracy for the semi-daily pollution maps could be slightly increased.
In [3], the same modelling approach based on generalized additive models is used. Here, the models have high temporal and spatial resolutions of 30 min and 10 m by 10 m. As a result, a strong time dependency of air pollution in terms of time of day, weekday and season is identified. Furthermore, a heavy impact of traffic and meteorological features on the air pollution is observed.

1.3 Approach

Although the LUR model introduced in the previous section is able to generate fine-grained pollution maps, this method has two important drawbacks. First, it

is only possible to create air pollution maps with high accuracy at about semi-daily resolution. Since pollution concentration can change heavily throughout one day, it would be helpful to have accurate maps with at least hourly temporal resolution. A second drawback is that the model outputs no uncertainty information about the inferred UFP concentration. Providing such an uncertainty measure is important for a lot of applications. For a weather forecasting model for example, an uncertainty value for the predictions provides a measure of how much the weather forecast can be trusted. In the case of air pollution inference, having a confidence value of predictions for out-of-distribution examples is important as the performance at locations where no measurements are available cannot be evaluated otherwise. By generating uncertainty maps, the generalization abilities of the model can be investigated and regions where more data are needed for more accurate predictions are discovered.

The goals of this thesis are to improve the temporal resolution of air pollution inference and to propose a model for uncertainty estimation of the inferred air pollution concentration.
In order to improve the temporal resolution compared to the LUR model, weather and traffic features are aggregated to capture important time-dependent dynamics (Subsection 3.1.2 and Subsection 3.1.3). Furthermore, a neural network approach (Subsection 3.2.2) is proposed and tested on the OpenSense dataset. With the new features and the neural network, a higher temporal resolution of the pollution maps is expected.
To provide an uncertainty measure, an approach is required where instead of a point estimation, a probability distribution is modelled. Traditionally, neural networks provide no measure of uncertainty. However, there are methods that adapt standard networks for approximating the probability distribution of a prediction. In this work, two methods for providing an uncertainty measure for a neural network are introduced (Section 2.2) and tested for air pollution inference.

In Chapter 2, related work on both the use of neural networks for generating pollution maps and how to get an uncertainty measure using neural networks is presented. Chapter 3 describes the used features and the proposed models, which are evaluated in Chapter 4. Finally, Chapter 5 concludes this thesis.

Chapter 2

Related work

In this chapter, related work about air quality inference (Section 2.1) and uncertainty estimation (Section 2.2) using neural networks and ensemble methods is presented.

2.1 Deep Learning and ensemble approaches for air quality inference

There are various linear and non-linear statistical approaches to tackle the problem of air quality estimation. Neural networks have shown to perform well for generating pollution maps with high spatio-temporal resolution. There are numerous approaches for air pollution forecasting [4, 5, 6, 7] and inference [8, 9, 10] using a neural network model.
In this section, some of these air pollution inference approaches are introduced.

2.1.1 U-Air: When Urban Air Quality Inference Meets Big Data

In U-Air [8], air quality information (SO_2, NO_2, $PM_{2.5}$ and PM_{10}) of Beijing is inferred using a semi-supervised learning approach based on a co-training framework. Two separate classifiers are used for the classification of air pollution levels: an artificial neural network (ANN) that captures the spatial dependencies and a linear-chain conditional random field (CRF) for modelling the temporal dependencies.

In Table 2.1, an overview of the features used is given. Road-network-related features include features extracted from the street network like the total length of highways and other road segments as well as the number of intersections of streets within a grid cell. Point of interest (POI) related features indicate the land use, function and the traffic patterns of a region.
The meteorological features used in U-Air are temperature, humidity, barometer pressure, wind speed and weather (such as cloudy, foggy, rainy, sunny and

Spatially-related features	Temporally-related features
Road-network-related features POI-related features	Meteorological features Traffic-related features Human mobility features
→ ANN	→ CRF

Table 2.1: Features used by U-Air

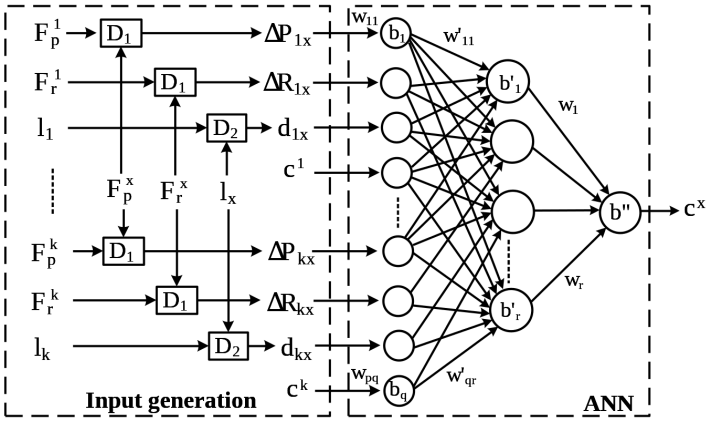


Figure 2.1: Neural network for the spatial classifier as proposed in [8].

snowy). Traffic-related features include information about the speed of vehicles traversing a grid and human mobility features represent information about the number of people arriving or departing from a grid.

The temporal and the spatial classifier are trained on the two separated sets of features and the probability scores of both classifiers are combined in the inference step.

In Figure 2.1, the structure of the spatial classifier is showed, where F_p^k indicate the POI features and F_r^k the road-network features at a certain location. For generating inputs of the artificial neural network, the correlation between the POI features (ΔP_{kx}) and road-network features (ΔR_{kx}) of some already labeled grids and the grid that is to be labeled as well as a distance measure (d_{kx}) between the two grids are computed. These correlations and distances are then feeded with the true labels (c^k) into a one-layer neural network to get an estimation of the label of the unknown grid cell.

U-Air outperformed other classical air pollutant dispersion models by far. Especially the modelling of spatial and temporal dependencies with a separate classifier and using co-training does a good job. This approach showed the importance of having features with a high spatial resolution as well as features with

a high temporal resolution. Since the LUR model lacks timely resolved variables, more features are aggregated in order to model the temporal dependency of air pollution. Moreover, the good performance of the ANN in U-Air motivates the usage of a similar structure in this work.

2.1.2 Extending Urban Air Quality Maps Beyond the Coverage of a Mobile Sensor Network: Data Sources, Methods, and Performance Evaluation

In [9], air quality of Lausanne is inferred using a deep learning framework that is based on autoencoders. Unlike in U-Air, spatial and temporal dependencies are learned simultaneously.

The features used in this model are: land-use features (such as altitude, population and industry), traffic features (such as daily and hourly mean charges) and some weather and pollutant data recorded by two static monitoring stations.

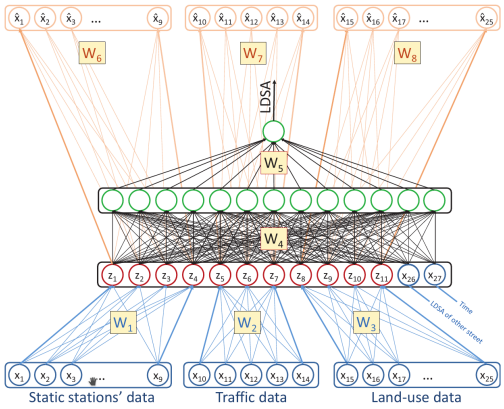


Figure 2.2: Neural network as proposed in [9].

In Figure 2.2, the neural network used for estimating Lung-Deposited Surface Area (LDSA) is shown. An autoencoder (W_1, W_2, W_3, W_6, W_7, W_8) is first trained on these features to extract the informative features of the data and to lower the input dimensions. Together with the time of the day and the air pollution measurement of the most similar street segment, the outputs of the autoencoder are feeded into a one layer neural network (W_4 and W_5).

As a result, this deep learning model outperformed two log-linear regression models which have already delivered acceptable results. Nevertheless, the computational complexity of the neural network approach was much higher than the compared models. This paper approves the usability of a neural network for air pollution inference.

2.1.3 Spatially Fine-grained Urban Air Quality Estimation Using Ensemble Semi-supervised Learning and Pruning

In [11], air pollution concentration is inferred by using an ensemble semi-supervised learning approach. Multiple classifiers are generated from the original dataset and then retrained by an interative co-training process.

This approach uses traffic-related features, road-network-related, POI-related, check-in features and nearby monitoring-related features. Check-in features model human mobility by counting the number of check-ins of people in social networking services. Nearby monitoring-related features take the air quality and correlation of features of nearby measuring stations into account.

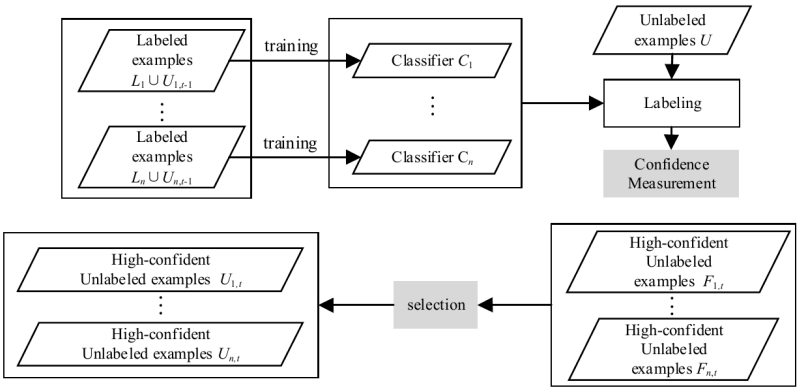


Figure 2.3: Ensemble approach as proposed in [9].

In Figure 2.3, the procedure of the semi-supervised ensemble algorithm is described. Multiple classifiers are first trained on bootstrap samples (L_i) of the original dataset and a label and confidence measure is assigned to the unlabeled examples. High-confident examples are added to the training dataset and the classifiers are trained again. This iterative process is repeated until a label is assigned to all the unlabeled examples.

This method outperformed different air quality estimation methods like Gaussian process regression [12] and U-Air [8]. The effect of using multiple regressors and combine them as an ensemble prediction is evaluated in this work for air pollution inference as well as uncertainty estimation.

2.2 Methods for uncertainty estimation

Measuring the predictive uncertainty of deep neural networks is a key factor for improving the accuracy and evaluating the estimation quality. One method of approximating the uncertainty are Bayesian neural networks. These networks

require large changes to the training procedure and they are computationally way more expensive than standard neural networks. However, there are approaches for uncertainty estimation in non-Bayesian neural networks which achieve similar or better performance than Bayesian NNs. In the next subsections, a method that is based on using an ensemble of neural networks (Subsection 2.2.1) and a method based on using dropout (Subsection 2.2.2) is presented.

2.2.1 Ensemble approach

In [13], a predictive uncertainty estimation method based on deep ensembles is described. A neural network used for regression usually outputs one single value $\mu(x)$. To capture uncertainty, the observed value is treated as sample from a Gaussian distribution with predicted mean $\mu(x)$ and variance $\sigma^2(x)$. Therefore, the neural network is changed in order to output mean and variance of the distribution [14].

A neural network that models a probabilistic predictive distribution $p_\theta(y|\mathbf{x})$ is used. In the context of air pollution inference, y would be the air pollution concentration, $\mathbf{x}$ a feature vector and θ indicates the parameters of the neural network. This approach makes use of three key techniques:

1. proper scoring rule as training criterion
2. adversarial training
3. ensemble training

Proper scoring rules
A scoring rule measures the performance of probabilistic predictions by assigning a numerical score to a predictive distribution $p_\theta(y|\mathbf{x})$. For a proper scoring rule, the score is only maximized if the predicted distribution equals the true distribution. In this paper, the score function used is $\log p_\theta(y|\mathbf{x})$, which is a proper scoring rule. Therefore, the neural network is built in order to minimize the negative log-likelihood criterion of a Gaussian distribution.

Adversarial training
Adversarial training is used to smooth the predictive distributions, which results in an improved robustness of the regressor. The idea here is to compute the direction of the feature vector where the loss is likely to increase and add perturbated samples which are close to the original training samples to the dataset for minimizing the loss in every iteration.

Ensemble training
In this work, a randomization-based ensemble approach is used, meaning that the regressors are trained in parallel. Because the usage of bagging deteriorated the performance, the entire training set is used to train each regressor. The ensemble

is treated as mixture of Gaussian distributions resulting in the following mean and variance of the mixture:

$$\mu(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\mu_{\theta_m}(\mathbf{x}),$$
$$\sigma^2(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}(\sigma^2_{\theta_m}(\mathbf{x}) + \mu^2_{\theta_m}(\mathbf{x})) - \mu^2(\mathbf{x}).$$

The ensemble approach performs as well as Bayesian neural networks, requires little hyperparameter tuning and is well suited for large scale distributed computation. In this work, the ensemble method for modelling predictive distribution of a neural network is applied for the problem of air pollution inference (Subsection 3.2.3).

2.2.2 Dropout approach

In [15], a method for estimating predictive uncertainty that is based on dropout is described. As for the ensemble method, the predictions are modelled as probability distributions instead of point estimations. The key techniques used in this approach are:

1. tunable proper scoring rule
2. dropout training

Proper scoring rule
Like in the deep ensemble approach (Subsection 2.2.1), the usage of a proper scoring rule is inevitable for training an effective neural network. In RDeepSense, the weighted sum of negative log-likelihood and mean square error is used as a loss function. This tunability should avoid the effect over- and underestimation of the predictive uncertainty.

Dropout
Dropout is often used in fully-connected neural networks as a regularization method to avoid feature co-adapting and overfitting. In each iteration of training, a given rate of weights are ignored in each layer of the neural network. These dropout operations convert a traditional neural network into a statistical model, which is mathematically proved in [15].

In order to approximate the predictive distribution $p(y|\mathbf{x})$, Monte Carlo estimation is used. For the case of regression, this results in an average of Gaussian distributions which can be approximated by one Gaussian distribution with following mean and variance:

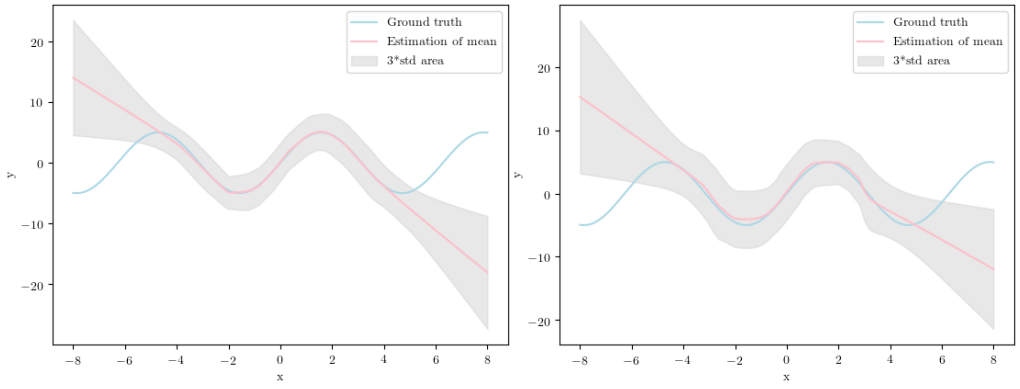
$$\mu(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\mu_m(\mathbf{x}),$$
$$\sigma^2(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}(\sigma_m^2(\mathbf{x}) + \mu_m^2(\mathbf{x})) - \mu^2(\mathbf{x}).$$

In RDeepSense, a recipe without the usage of Monte Carlo estimation is introduced. During test time, all the weights are multiplied by the dropout rate. Like this, the neural network has to be run only once which is especially benefitial for usage in mobile applications. This efficient variant performed better in various tests even though it is mathematically not equivalent to the Monte Carlo estimation.

The dropout method outperforms state-of-the-art baselines on the quality of uncertainty estimations. It is way more effective in terms of energy consumption than the ensemble method, but has the disadvantage that it is only applicable on fully-connected neural networks. This approach is applied on the problem of air pollution inference and is compared to the ensemble method regarding the quality of uncertainty in this work.

2.2.3 Comparison

For demonstration and comparison of the two methods, a one-dimensional toy regression dataset is generated. Training samples are drawn from $y = 5\sin(x) + \epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$. For the training set, samples with $x \in [-8, 8]$ are generated, for the testset, samples are in the range $x \in [-16, 16]$. In Figure 2.4, the two methods are compared. Both models are able to estimate the mean in the range where training data is available. However, in the range where no data is used for training, the mean deviates a lot from the ground truth and the predicted standard deviation corresponds to this uncertainty for both the dropout and ensemble approach.



(a) Ensemble method. (b) Dropout method.

Figure 2.4: Comparison of predictive uncertainty estimation of the ensemble and dropout method on a toy dataset.

Chapter 3

Data aggregation and Model

In this chapter, the applied features, UFP data (Section 3.1) and models (Section 3.2) are described.
The same land use features that are used for the land-use regression model [2] are used in this thesis (Subsection 3.1.1). These features are averaged values that vary only spatially and include no temporal dependencies. In order to capture the temporal dependencies of air pollution concentration, new traffic and weather variables that are dynamic in time are aggregated (Subsection 3.1.2 and Subsection 3.1.3). The aggregation of ultra fine particle (UFP) measurements is described in Subsection 3.1.4.
In this thesis, long short-term memory networks (Subsection 3.2.1) and fully-connected neural networks (Subsection 3.2.2) are used for predicting air pollution concentration based on the features and UFP data. Moreover, this model is changed such that it is able to provide an uncertainty measure of its predictions (Subsection 3.2.3).

3.1 Features and UFP data

A high quality of air pollution measurements and the explanatory variables is a requirement for a successful air pollution inference. In this thesis, three types of features are used: the static land use and traffic data obtained from the LUR model (F_1), dynamic traffic features with spatial and temporal resolution (F_2) and timely resoluted weather features (F_3).

3.1.1 F_1: Static land use and traffic data

In Table 3.1, the features that are used for the land-use regression model introduced in Section 1.2 are listed. As in Hasenfratz et al. [2], a high correlation ($R^2 > 0.6$) between population, floorlevel and heating is observed. Replacing these variables by only one of them should not result in a decreased performance.

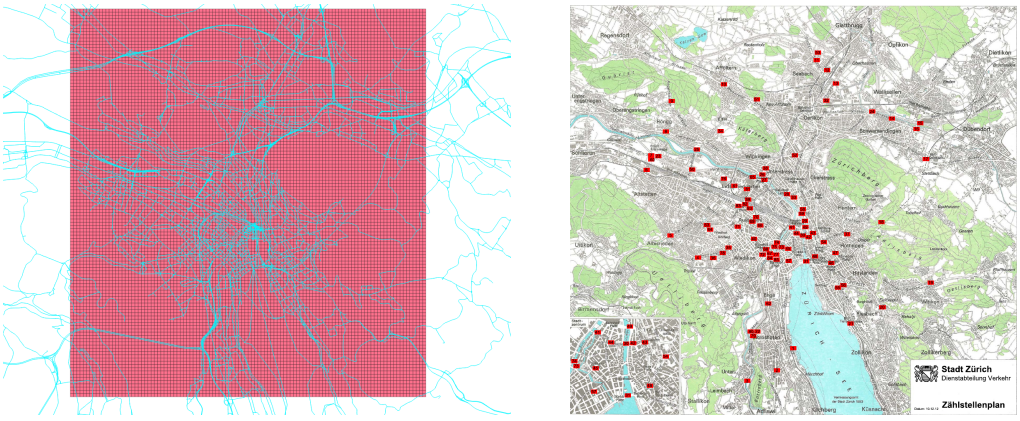
In the land-use regression model, two variables describing the traffic volume are used. However, these values only describe a mean value and are not time-resolved. To improve the model, time-resolved traffic features are needed. In the next subsection, the extraction of some new traffic features is described.

Variable [unit]	**Variable [unit]**
Population [inhabitants/ha]	Industry [industry buildings/ha]
Building height [floor levels/ha]	Heating [oil and gas heatings/ha]
Terrain elevation [average m/ha]	Road type [busiest road type/ha]
Distance to next road [m]	Distance to next large road [m]
Terrain slope [average degree/ha]	Terrain aspect [average degree/ha]
Traffic volume [vehicles per day/ha]	Distance to next traffic signal [m]

Table 3.1: Spatially resolved land use and traffic features [2].

3.1.2 F_2: Dynamic traffic features

Traffic is one of the most significant sources that produce air pollutants. Therefore, the quality of a model depends strongly on the exactness and resolution of traffic features. In order to catch the spatial and timely dependencies of the traffic, two data sources are used: the cantonal average daily traffic model of Zurich and the data of several counting stations in the city of Zurich.



(a) Traffic network of Zurich and the grid considered in this thesis.

(b) Counting stations in the city of Zurich.

Figure 3.1: The two data sources used for extracting traffic features.

The cantonal average daily traffic model of the Canton Zurich yields a precise model of daily traffic and public transport volume. For this work, the average

daily traffic model (DTV[1]) and the daily traffic model for an average working day (DWV[2]) is used. Figure 3.1a shows the traffic network and a grid that includes 13200 cells of size 100m x 100m. Two numbers (one for each direction) are assigned to each segment i. For street segments, this number represents the number of vehicles ($N_v^{(i)}$), while in public transport segments, it describes the number of commuters ($N_c^{(i)}$). In each grid cell, different features are extracted for both the DTV and DWV:

- Sum of the length of a street segment multiplied by $N_v^{(i)}$ (dtv1/dwv1)
- Maximum $N_v^{(i)}$ over all street segments (dtv2/dwv2)
- Incoming $N_v^{(i)}$ (dtv3/dwv3)
- Sum of the length of a public transport segment multiplied by $N_c^{(i)}$ (dtv4/dwv4)
- Maximum $N_c^{(i)}$ over all street segments (dtv5/dwv5)
- Incoming $N_c^{(i)}$ (dtv6/dwv6)

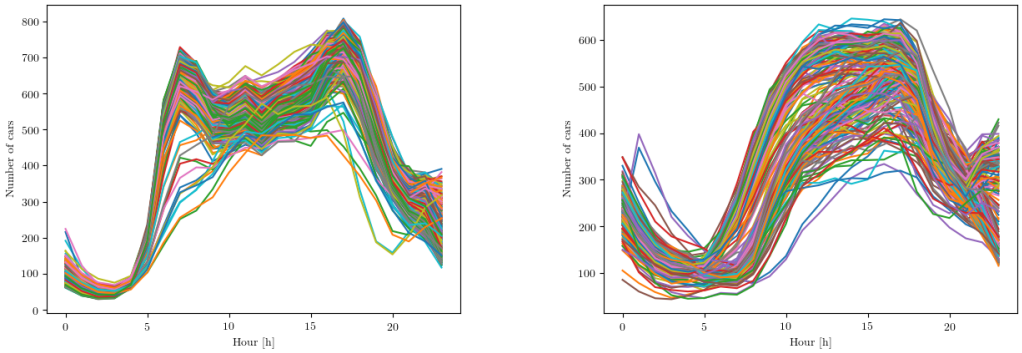
In addition, the length of all segments (len_streets) in a cell is summed and used as the last of the 13 features in total.

As a second source of traffic information, the data of 85 vehicle counting stations in Zurich is used (Figure 3.1b). The mean value of all these stations is computed in hourly resolution for 2012 and 2013. Figure 3.2 shows a comparison of the average traffic counts for a working day and a holiday or weekend day. As can be expected, more vehicles are counted on working days in general. Furthermore, traffic load peaks in the morning and evening for working days, whereas there is only one peak in non-working days.
In addition to the mean of the counting stations, a second feature is extracted that combines the two data sources. The idea is to get a value for the number of cars passing each cell for a specific day and hour. To reach that, the counting station values are averaged over the year and multiplied by the incoming number of cars in each cell from the DTV or DWV models.

Table 3.2 shows an overview of the aggregated traffic features. 'dtv*' and 'dwv*' describe the features extracted from the DTV and DWV model, 'cars1' is the combination of DTV/DWV and the counting stations and 'cars2' is the mean extracted from all the counting stations.

In Figure 3.3, the correlation matrix of all the traffic features is shown, where 'traffic' and 'traffic_tot' are the traffic features from the LUR model. The features 1-3 from DTV and DWV, which represent the features extracted from the street

[1]DTV: Durchschnittlicher Tagesverkehr
[2]DWV: Durchschnittlicher Werktagesverkehr



(a) Working day.

(b) Weekend or holiday.

Figure 3.2: Hourly mean of all the counting stations for a working day (left) or a weekend or holiday (right).

Variable [unit]	**Variable [unit]**
dtv1/dwv1 [vehicles·m/ha]	dtv4/dwv4 [commuters·m/ha]
dtv2/dwv2 [vehicles/ha]	dtv5/dwv5 [commuters/ha]
dtv3/dwv3 [vehicles/ha]	dtv6/dwv6 [commuters/ha]
len_streets [m/ha]	cars_1 [vehicles/ha]
cars_2 [vehicles]	

Table 3.2: Traffic features extracted from DTV, DWV and the counting stations. 'cars_1' is spatially and timely resolved, 'cars_2' is only timely resolved and the others only spatially.

segments described above, highly correlate to each other and 'cars_1'. Similarly, features dtv4/dwv4 to dtv6/dwv6 from these models also highly correlate to each other, as they all represent some public transport feature. The effect of replacing the high correlated features by only one of them is examined later on.

3.1.3 F_3: Dynamic weather features

The concentration of air pollutants is highly influenced by meteorology. There even are air quality forecasting models that only include weather data (e.g. [6]). Therefore, it is inevitable to include meteorological features in order to push the temporal resolution of air pollution inference.

The weather data of two meteorological stations located at Mythenquai and Tiefenbrunnen are used. For temperature, humidity, wind chill, air pressure and dew point, the mean between the two stations is computed. Because the data of Mythenquai showed unrealistic wind measurements in some time ranges, the data of Tiefenbrunnen is taken here. Rain and global radiance is only measured

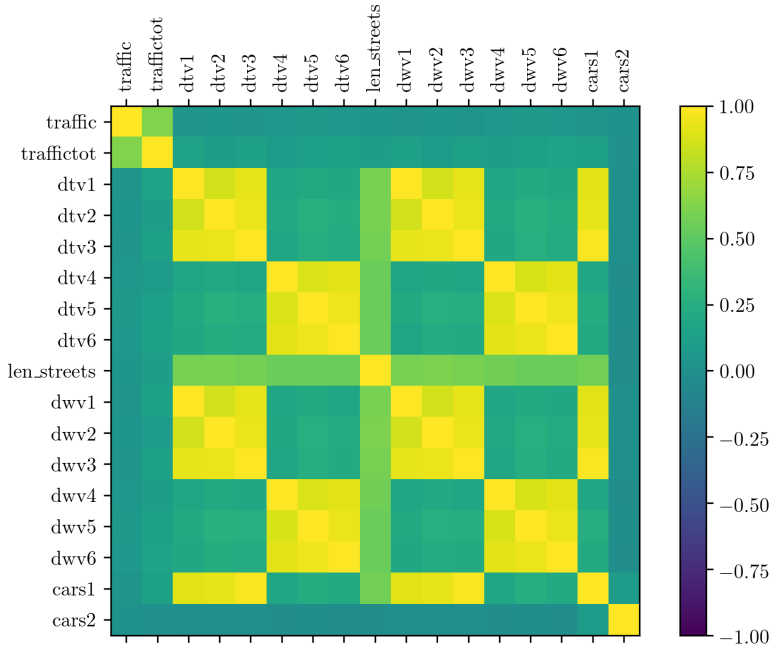


Figure 3.3: Correlation matrix of the traffic features.

in Mythenquai. In total, 11 weather features (Table 3.3) are extracted for every 10 minutes from April 2012 to March 2013.

Variable [unit]	**Variable [unit]**
Temperature [°C]	Humidity [%]
Wind gust [m/s]	Wind velocity [m/s]
Wind strength [bft]	Wind direction [°]
Wind chill [°C]	Air pressure [hPa]
Dew point [°C]	Rain [mm]
Global radiance [W/m^2]	

Table 3.3: Timely resolved (10 min) weather features.

In Figure 3.4, the correlations between each meteorological feature is shown. As expected, all the features describing a temperature (temperature of the air, wind chill and dew point) and all the wind features (wind gust, wind velocity and wind strength) highly correlate with each other. In Subsection 4.2.2, the effect of replacing these high-correlating features with only one of them is examined.

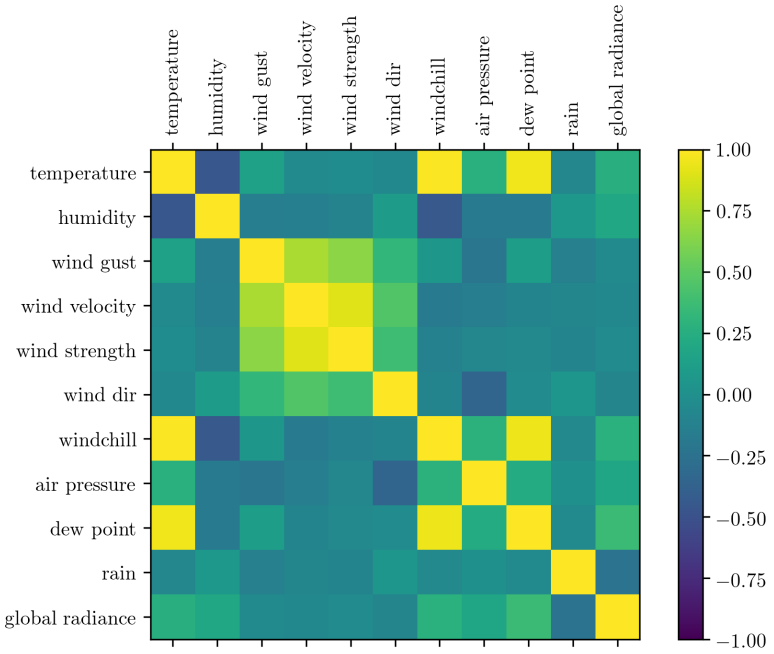


Figure 3.4: Correlation matrix of the weather features.

3.1.4 UFP data

In most countries, air quality regulations only consider the mass of particulate matter with a diameter of less than 10 μm (PM_{10}) and 2.5μm ($PM_{2.5}$). However, studies have shown that ultrafine particles (particulate matter with diameter less than 0.1 μm) are more toxic than larger particles [16]. In order to monitor concentrations of UFP, a mobile measurement system consisting of ten sensors on top of trams collected over 75 million UFP measurements over a time period of three years from April 2012 to May 2015.
UFP concentration is sampled every 50 ms and is aggregated to one sample for every 5 s in order to reduce the amount of tramsmitted data. The timestamped and geo-tagged measurements are calibrated and filtered in order to remove faulty and unreliable measurements. A high spatial resolution is guaranteed by dividing the city into grid cells of 100m x 100m and a good spatial coverage is reached by the UFP measurements (Figure 3.5).
In this thesis, air pollution prediction is examined for various temporal resolutions. For yearly to biweekly resolution, land use features F_1 are assigned to the averaged pollution concentration according to the location of the measurements. The data from April 2012 to April 2013 is used for air pollution prediction of daily to hourly resolution. For this purpose, UFP data and the timely dynamic weather and traffic features are averaged over the given time range and all presented features are allocated to each averaged UFP concentration according to

the time and location.

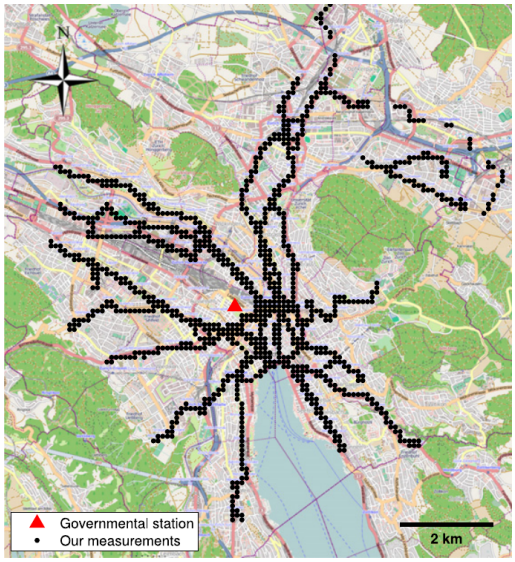


Figure 3.5: Spatial coverage of Zurich by the mobile sensor network. [2]

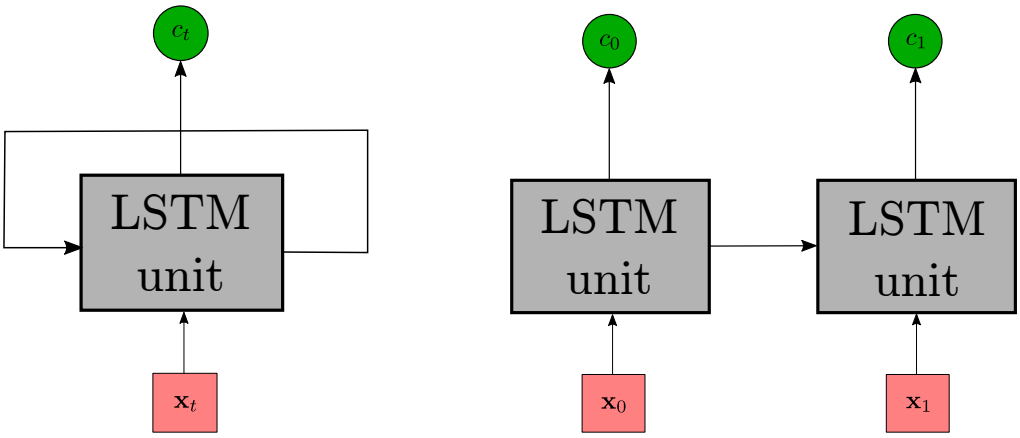
3.2 Modeling methods

Artificial neural networks have shown to be usable for different tasks like speech recognition, computer vision or medical diagnosis. They are able to approximate an arbitrary function by learning from data and are often used to represent a mathematical model. As introduced in 2.1, neural networks are also successfully used for air pollution inference. In the next subsections, a short introduction in long short-term memory networks (Subsection 3.2.1) and fully-connected neural networks (Subsections 3.2.2 and 3.2.3) and their usage in this project is presented.

3.2.1 Long short-term memory

Recurrent neural networks (RNN) are a family of neural network that address the issue of processing sequential data. Unlike traditional feedforward neural networks, RNNs are able to learn a temporal context of input sequences. This is achieved by using a loop structure (Figure 3.6a) where the output at a timestep is used as an input to the next timestep.

Long short-term memory (LSTM) networks are a special kind of RNNs that work better for most tasks as they learn much faster and solve the problem of long-term dependencies [17]. They are widely used for numerous applications like robot control, speech recognition or music compostion.



(a) Loop structure of LSTM.

(b) Unrolled LSTM for two timesteps.

Figure 3.6: Loop structure and unrolled LSTM network.

In Figure 3.6b, the LSTM is shown when unrolled for two timesteps. This network first takes $\mathbf{x}_0$ from the sequence and outputs c_0, which is also the input to the next time step together with $\mathbf{x}_1$. This process makes sure to remember the temporal context of the sequences and can be continued for an arbitrarily number of timesteps. However, only sequences of length two are used in this project as the grid cells where longer time sequences of UFP measurements are available is limited.

Two tests using an LSTM network are carried out. First, the usability of LSTMs for air pollution forecasting based on the weather measurements is checked. Given the meteorological conditions as well as the pollution concentration for the prior day, the pollution of the next day is predicted by the LSTM. In a second experiment, the performance and predictions of using an LSTM with all features as an input is compared to using a standard neural network with one hidden layer (Subsection 4.2.1).

3.2.2 Neural network for air pollution prediction

Neural networks often consist of several layers with multiple neurons that are connected by weights. If all the neurons from each layer are connected to each neuron from the previous layer, the network is called fully-connected. In Figure 3.7, an example of a fully-connected neural network with two hidden layers, is shown. A similar structure is used for air pollution inference in this project, where F_1 - F_3 are the different feature types and c is the inferred pollution concentration.

These networks are called feedforward neural networks or multilayer perceptrons (MLPs), because information flows through the network from the input to

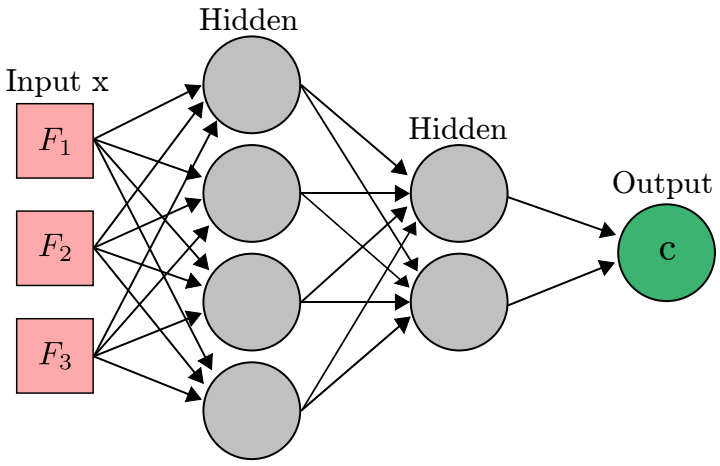


Figure 3.7: Fully-connected neural network with two hidden layers used to model the dependencies of features F_1 - F_3 and the pollution concentration c.

the output [18]. A function $f(\mathbf{x})$ is approximated by chaining different functions. For the example of the neural network with two hidden layers, $f(\mathbf{x})$ is composed of three functions ($f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$). Here, $f^{(1)}$ is called the first layer, $f^{(2)}$ the second layer and $f^{(3)}$ the output layer. In each layer, an activation function is applied to the weighted inputs:

$$\mathbf{y} = \text{activation}(W^T\mathbf{u} + \mathbf{bias}),$$

where activation is an element-wise activation function, $W \in \mathbb{R}^{n\times m}$ is the weight matrix, $\mathbf{u} \in \mathbb{R}^n$ the input and $\mathbf{y} \in \mathbb{R}^m$ the output from a layer. When training the network, the weights and biases are adapted such that a specified cost function is minimized for the training examples. A popular cost function that is often used in neural networks is the mean squared error $\frac{1}{n}\sum_{i=1}^{n}(c_i - \hat{c}_i)^2$, where c_i is a training sample and $\hat{c}_i$ are the estimations by the network. To minimize this cost function, it is essential to numerically compute the gradient of the function, which is done by the simple and popular back-propagation algorithm [18]. In order for this back-propagation algorithm to converge, the input and output data is standardized to have zero mean and unit variance.

In the case of air pollution inference, the goal is to find a function $c = f(\mathbf{x})$ which models the dependencies of the explanatory variables $\mathbf{x}$ (land use, traffic and weather features) and the concentration c of air pollutants. The performance of neural networks with one, two or three hidden layers and their combination as an ensemble is evaluated in Section 4.2.2.

3.2.3 Neural networks for probability distribution estimation

In order to get an uncertainty measure, the observed pollution concentration is treated as a sample from a Gaussian distribution with mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$ and the searched function is changed to a probabilistic predictive distribution $p_\theta(c|\mathbf{x})$. Instead of having only one output of the neural network, the structure is altered such that it outputs both mean and variance of the Gaussian distribution [19] (Figure 3.8). To enforce a positivity constraint on the variance, this output is passed through the softplus function $\log(1 + \exp(\cdot))$.
In this thesis, the deep ensemble [13] and dropout [15] approach (Subsections 2.2.1 and 2.2.2) are applied on air pollution inference, where the outputted variance $\sigma^2(\mathbf{x})$ is used as an uncertainty measure.

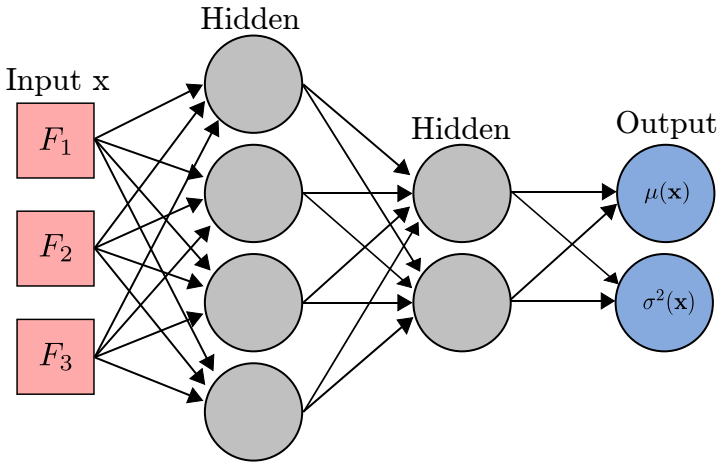


Figure 3.8: Fully-connected neural network with two hidden layers used to model a predictive distribution $p_\theta(c|\mathbf{x})$.

The results of the uncertainty prediction methods based on ensemble and dropout methods are presented in Subsection 4.2.3. Furthermore, the empirical variance of the outputted means of the ensemble members is tested as a third uncertainty measure.

Chapter 4

Evaluation

In this chapter, the performance of neural networks for air pollution prediction and the reliability of uncertainty estimations of proposed models in Section 3.2 are evaluated. Therefore, the quality of predicted UFP concentrations is tested on the OpenSense dataset (Subsection 4.2.1 and 4.2.2) and is compared to the genenerative additive model. The reliability of predictions at locations where no measurements are obtained is examined by generating uncertainty maps (Subsection 4.2.3).
The used metrics and methods for evaluating the quality of predicted air pollution and uncertainty measures are presented in Section 4.1 and the results are shown in Section 4.2.

4.1 Evaluation metrics and methods

Root-mean-square error, normalized root-mean-square error, coefficient of determination and factor of 2 measure are used for evaluating the performance of air pollution inference (Subsection 4.1.1). For illustrating the quality of uncertainty estimations, calibration curves, an evaluation in bins and rank histograms are used (Subsection 4.1.2).

4.1.1 Metrics for the quality of inference

Four standard metrics are used to evaluate the quality of the predicted pollution concentrations $\hat{c}_i$ compared to the measured values c_i. In order to get realiable evalution metrics, a 10-fold cross-validation is applied, which is also described in this subsection.

Root-mean-square error (RMSE) and normalized root-mean-square error (NRMSE)

Root-mean-square error is a measure for the differences between predicted and measured values. The lower the RMSE is, the more accurate is the model. However, this metric does not allow for a comparison between different datasets. That's why the normalized root-mean-square error (NRMSE) is computed in addition to the RMSE. The RMSE normalized by the standard deviation of the measurements (σ_c) yields the normalized RMSE, where $\bar{c}$ is the mean of the measured values.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(c_i - \hat{c}_i)^2}$$

$$\text{NRMSE} = \frac{\text{RMSE}}{\sigma_c} = \frac{\sqrt{\sum_{i=1}^{N}(c_i - \hat{c}_i)^2}}{\sqrt{\sum_{i=1}^{N}(c_i - \bar{c})^2}}$$

Coefficient of determination (R^2)

The coefficient of determination represents how well the predictions of a regression fit the measured data. The closer this value gets to 1, the better is the fitting. R^2 is defined by

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(c_i - \hat{c}_i)^2}{\sum_{i=1}^{N}(c_i - \bar{c})^2}$$

Factor of 2 measure (FAC2)

Factor of 2 computes the fraction of the test data that satisfies:

$$0.5 \leq \frac{\hat{c}_i}{c_i} \leq 2.0.$$

This corresponds to the fraction of the data that lie in the factor two area of a scatter plot of the predictions against measurements.

10-fold cross-validation (CV)

As suggested in [2], a 10-fold cross-validation is performed for testing the model's ability to predict the air pollution concentration of new data. The original

dataset is randomly partitioned into 10 equally-sized subsets, where one of these subsets is retained for testing while the other 9 subsets are used as training data. This process is repeated until every subset is used as test data once. In this way, too optimistic results of only one simple train/test split can be avoided and the skill of a model on unseen data can be estimated reliably.

4.1.2 Methods for the quality of uncertainty estimation

For evaluating the quality of the uncertainty prediction models, three evaluation methods (calibration curves, rank histograms and evaluation in bins) are used.

Calibration curves

The first method is based on calibration curves, also called reliability diagrams [15, 20]. A good predictive model should be well calibrated, which means that the observed frequencies of samples that lie within a confidence interval should truthfully reflect this confidence interval (predicted frequency). In case of modelling a Gaussian distribution, 68.3% of the samples should lie within the standard deviation (68.3% confidence interval) of the predicted mean ($\mu \pm \sigma$) for example. Ideally, the observed frequencies perfectly matches the predicted frequencies, which results in a diagonal calibration curve.
For a set of confidence intervals $z = [10\%, 20\%, \cdots, 80\%, 90\%]$, the fraction of predictions where the true value lies within the bounds of the confidence interval is computed and plotted against the value z. By considering these calibration plots, it can be evaluated if predictions tend to over- or underestimate predictive uncertainty. In Figure 4.1, the calibration plots of two different predictive models are shown. If a model tends to overestimate uncertainties (too high predicted variances), more true values lie in lower confidence intervals which leads to calibration curves that lie above the diagonal. Similarly, a calibration curve that lies below the diagonal is a sign that a model tends to underestimate uncertainties (too low variances are predicted).

Evaluation in bins

For evaluating the relationship between predictive uncertainty and absolute error of a prediction, the samples are divided in ten bins of equal size. The samples are sorted in increasing order regarding absolute error. The 10% of the samples with the smallest absolute errors are placed in the first bin, the next 10% are placed in the second bin and so on. The variances are scaled such that they are in a range from 0 to 1 and the mean of these scaled variances is computed in each bin. To show the raise of uncertainty from each bin compared to bin 1, the computed mean of bin 1 is subtracted. Like this, the raise in variance with

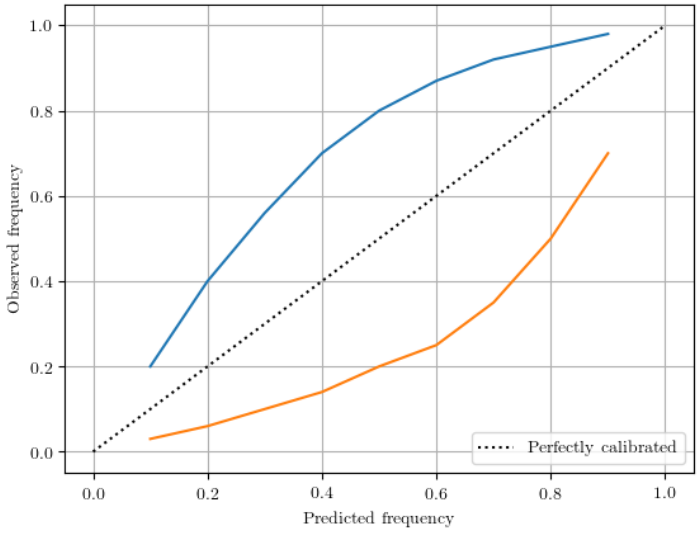


Figure 4.1: Example of calibration curves of overly confident predictions (orange) and overly cautious predictions (blue).

increasing absolute error is illustrated to see if samples with high absolute error also yield a high uncertainty value.

Rank histogram

Rank histograms are used to visualize the characteristics of an ensemble (Subsection 2.2.1). A rank indicates where a measured value lies compared to the predictions of the ensemble members. Rank $(k + 1)$ is assigned to a sample if k ensemble members predict a value lower than the true value. A rank histogram shows how many times each rank occurs. Ideally, this histogram is flat, meaning that the frequency of each rank is approximately the same. However, if there is a peak in low ranks, there is probably a high bias in the model as the observed value is too commonly lower than most of the ensemble predictions. If there are some low and some high biases or if the ensemble does not spread enough, more low and high ranks occur resulting in an U-shaped histogram.

4.2 Results

In this section, the results of the experiments are presented. First, the performance of an LSTM network for air pollution forecasting is compared with a

standard neural network (Subsection 4.2.1).
In Subsection 4.2.2, the fully-connected neural network model is compared to the land-use regression model in yearly to biweekly resolution. Furthermore, the neural network with the new weather and traffic features is tested for daily to hourly resolution and is compared to a generalized additive model (GAM) and a linear regression model for daily resolution. In addition, a sensitivity analysis on different feature sets is carried out.
The results of the tests with uncertainty estimations are presented in Subsection 4.2.3. Here, ensemble and dropout methods are compared and the spatial and temporal dependency as well as the dependency on features of the predicted uncertainties is examined.

4.2.1 Air pollution forecasting

The LSTM network for two timesteps introduced in Subsection 3.2.1 is tested using the daily means of pollution concentration and features. Pollution data is extracted from each grid cell where measurement data is available for two successive days. This dataset is scaled to a range between zero and one and is randomly splitted into training and test data. An LSTM with 10 units is trained for 1000 epochs and the batchsize is chosen as 128. The LSTM using only the weather features showed a decent performance for air pollution forecasting. (Table 4.1).
In a second test, the performance of LSTM with 10 units and a fully-connected neural network with one hidden layer of 52 nodes is tested using all of the features. Standard scaling is applied on features and pollution concentrations before training the networks for 1000 epochs. Both networks are evaluated on the same test set and it is observed that the absolute errors of the LSTM highly correlate to the errors of the NN. Samples predicted badly by the LSTM are also poorly predicted by the NN and vice versa. The final error metrics of the two tests are described in Table 4.1.

	RMSE	**NRMSE**	$\mathbf{R^2}$	**FAC2**
LSTM, weather features	3648.447	0.549	0.699	98.196
LSTM, all features	3455.779	0.518	0.731	98.261
NN, all features	3294.303	0.494	0.756	98.379

Table 4.1: Air pollution forecasting using an LSTM compared with NN approach.

A better performance is reached by a fully-connected neural network compared to a long short-term memory network. Furthermore, the LSTM is highly restricted in its usage since a prediction is only possible if a measurement from the day before is available. That is why only standard neural networks are used for further evaluation.

4.2.2 Air pollution inference

Yearly to biweekly resolution - comparison with LUR model

The neural network (NN) approach presented in Subsection 3.2.2 is first compared to the land-use regression model using only the static landuse data F_1 as input features. A small neural network of one hidden layer and 16 nodes is used and the comparison is carried out for yearly, seasonal, monthly and biweekly timescales.
In Figure 4.2, the results of a 10-fold CV are shown in a boxplot for the error metrics RMSE, R^2. The orange middle line of each boxplots shows the median of the error metrics and the box represents 50% of the data by extending from the lower quartile to the upper quartile. The length of the whiskers is restricted to 150% of the length of the box but ends if the maximal or minimal value lies in this range. Outliers are marked by circles. For FAC2, the mean of the cross-validation results is plotted because these values do not differ much and a boxplot is not illustrative.
The neural network model reached a similar performance as the land-use regression approach for the tested time scales. As for the LUR, the performance of the neural network model decreases when including more than 200 grid cells (Figure 4.3). This is due to the fact that the grid cells with a higher number of measurements are favored. Therefore, the number of cells with an unreliable average cell concentration increases when considering more cells as the calculated means are more and more based on a limited number of measurements.
Although the performance of these two models is similar, the generated maps differ significantly (Figure 4.4). The neural network predicts a very high pollution concentration at the corners of the considered region. This could be because of the small amount of training data (only 200 cells) and the little spread of features like the distance to roads in the training set. Since the measurements are taken by trams, the training data only covers points where the distance to roads is low. The neural network approch with that limited amount of training samples seems to struggle to predict the air pollution at locations where the input features differ a lot from features values in the training set. This inability of extrapolation is often observed with neural networks, which results in bad predictions outside of the range spanned by the training samples. As shown in Subsection 2.2.3, even a simple sine function could not be modeled outside the training range by a neural network model.

Daily resolution - comparison of different NN structures

In order to test the neural network model for higher temporal resolution, the timely resolved weather (F_2) and traffic (F_3) features are added. Here, all the features are allocated to the datapoints from April 2012 to March 2013 where

the daily mean is generated from over 50 samples. This dataset of 44469 samples is used for training and testing different structures of the neural network.
In Figure 4.5, the results of a 10-fold cross-validation of various neural networks are shown, where the numbers in brackets describe the number of nodes in each hidden layer of the network. All the models performed very similar regarding all the error metrics and making the network deeper did not necessarily result in a significantly improved performance.

Daily resolution - comparison of different models

Figure 4.6 shows the comparison of a single neural network consisting of 80 nodes in the first layer and 20 nodes in the second layer ([80, 20]) with a linear regression model, a generalized additive model (GAM) and an ensemble of neural network. A linear link function is chosen for GAM instead of the logarithmic function as used in the land use regression model (Section 1.2) because the standardized pollution concentrations can also include negative values. The ensemble of neural networks averages the predictions of 24 individually trained neural network models (8 models for 1, 2 or 3 hidden layers).
The evaluation plots (Figure 4.6) show that the ensemble of neural networks outperformed the single neural network model for all the evaluation metrics. A good performance is also reached by GAM, whereas the linear regression is too limited for modelling the non-linear dependencies of the features and air pollution concentration.
In Figure 4.7, the maps generated by the ensemble for one day in May, August, November and February are shown. It is observed that spatial differences are similar for each map, whereas predicted pollution levels are much higher in winter (February) compared to summer (August).

Daily resolution - importance of features

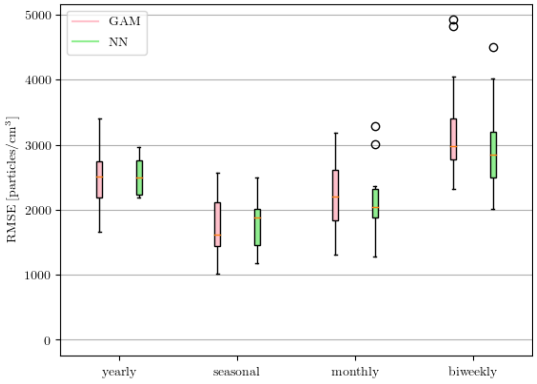
The performance of individual features and their combination is shown in Table 4.2. For F_4, the highly correlated features (Section 3.1) are removed and replaced by only one of them. A neural network with one hidden layer and 100 nodes is trained for each feature set and evaluated applying a 10-fold CV. The entries in the table correspond to the mean of the cross-validation results. They show that adding weather features results in a considerable improvement of the model. In general, adding a feature set into the model brings an improvement in each case, adding traffic features results in the lowest improvement however. Furthermore, performance is not decreased significantly when removing high correlated features.

Features	RMSE	NRMSE	R^2	FAC2
F_1	6058.926	0.894	0.2	92.469
F_2	4502.875	0.664	0.558	96.726
F_3	5919.972	0.874	0.237	92.957
$F_1 + F_2$	3227.12	0.476	0.773	98.309
$F_1 + F_3$	5869.625	0.866	0.25	93.587
$F_2 + F_3$	3734.033	0.551	0.696	97.888
$F_1 + F_2 + F_3$	3160.959	0.466	0.782	98.543
F_4	3185.045	0.47	0.779	98.426

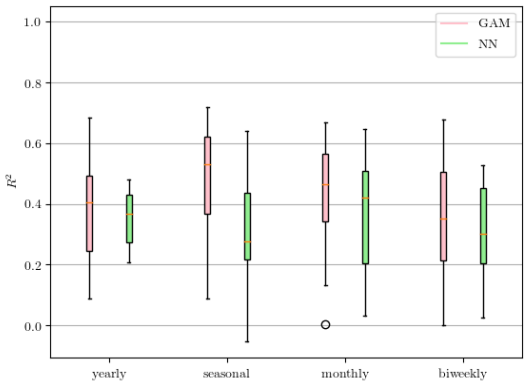
Table 4.2: Results related to land use (F_1), weather (F_2) and traffic features (F_3).

Daily to hourly resolution

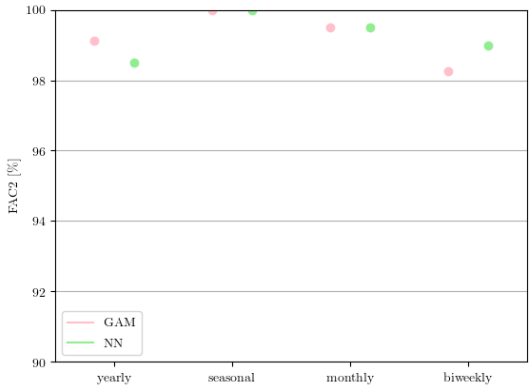
When increasing the timely resolution of the model, the problem of air pollution inference gets more complex. In order to test the performance of the neural network for up to hourly time scales, the pollution data and features are averaged over 18 hours (6-23), 9 hours (6-14, 15-23), 6 hours (6-11, 12-17, 18-23), 3 hours (6-8, ..., 21-23), 2 hours (6-7, ..., 22-23) and 1 hour. As for the daily dataset, only cells where pollution data is averaged over at least 50 cells is taken for training a neural network with one hidden layer of 60 nodes. The results of a 10-fold cross-validation is shown in Figure 4.8. The figure shows that performance decreases continuously when increasing the timely resolution. This is due to the lower amount of samples considered for aggregating a mean pollution and feature value, which gives a higher weight to outliers. Furthermore, it is difficult to estimate a pollution concentration of higher resolution because of increased fluctuations in the true values.



(a) RMSE.

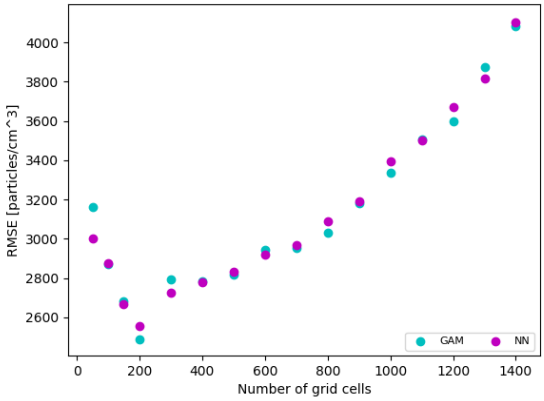


(b) R^2.

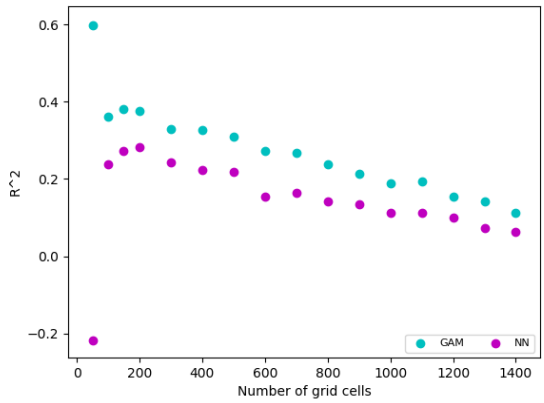


(c) Factor of 2.

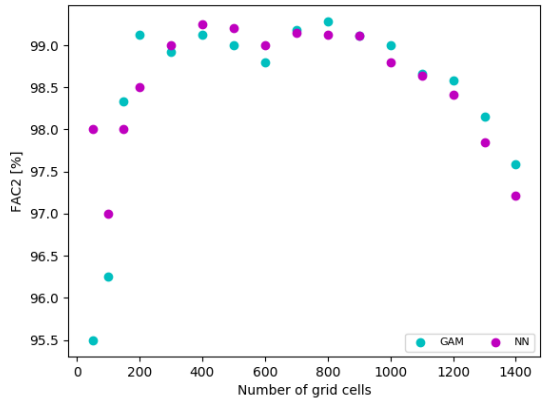
Figure 4.2: Comparison of the neural network approach with the generalized additive model.



(a) RMSE.

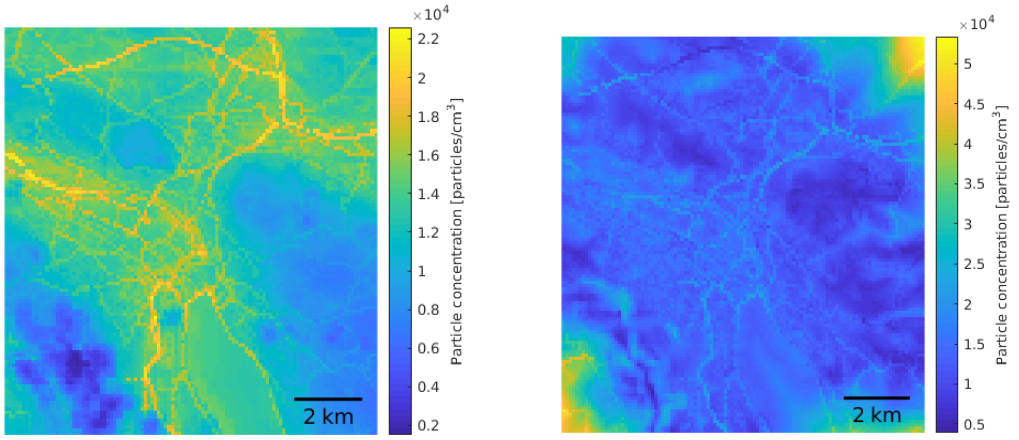


(b) R^2.



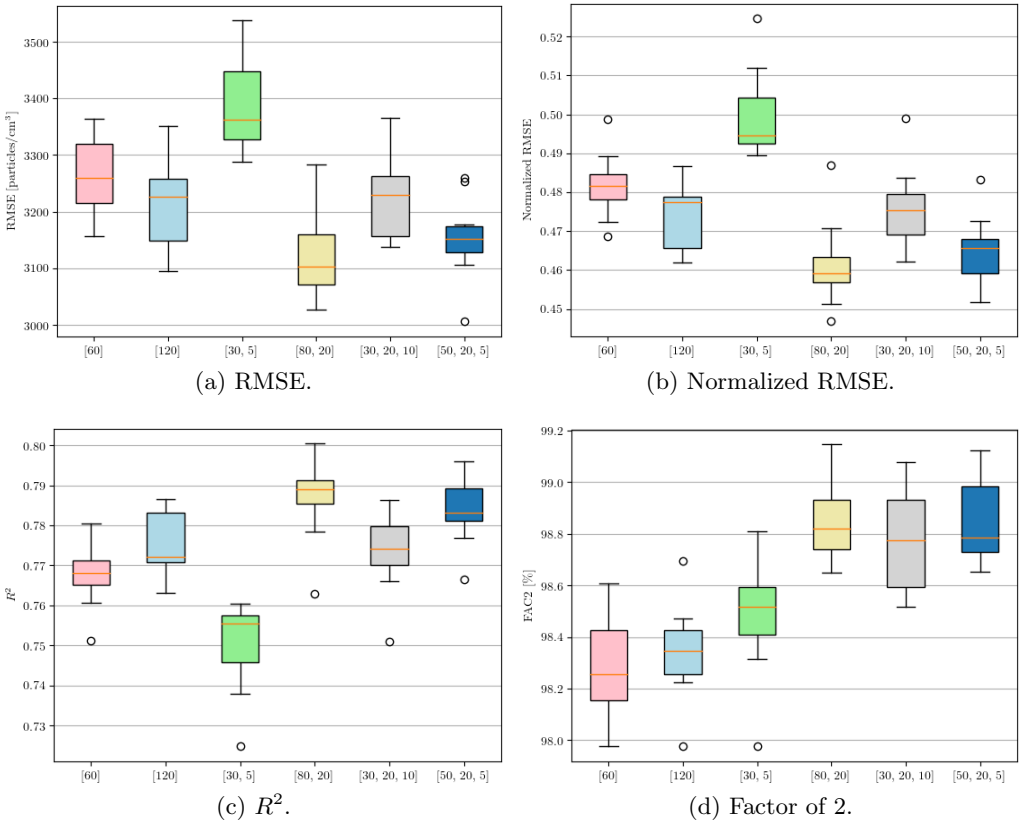
(c) Factor of 2.

Figure 4.3: Comparison of the yearly model with 50-1400 grid cells.



(a) GAM. (b) NN.

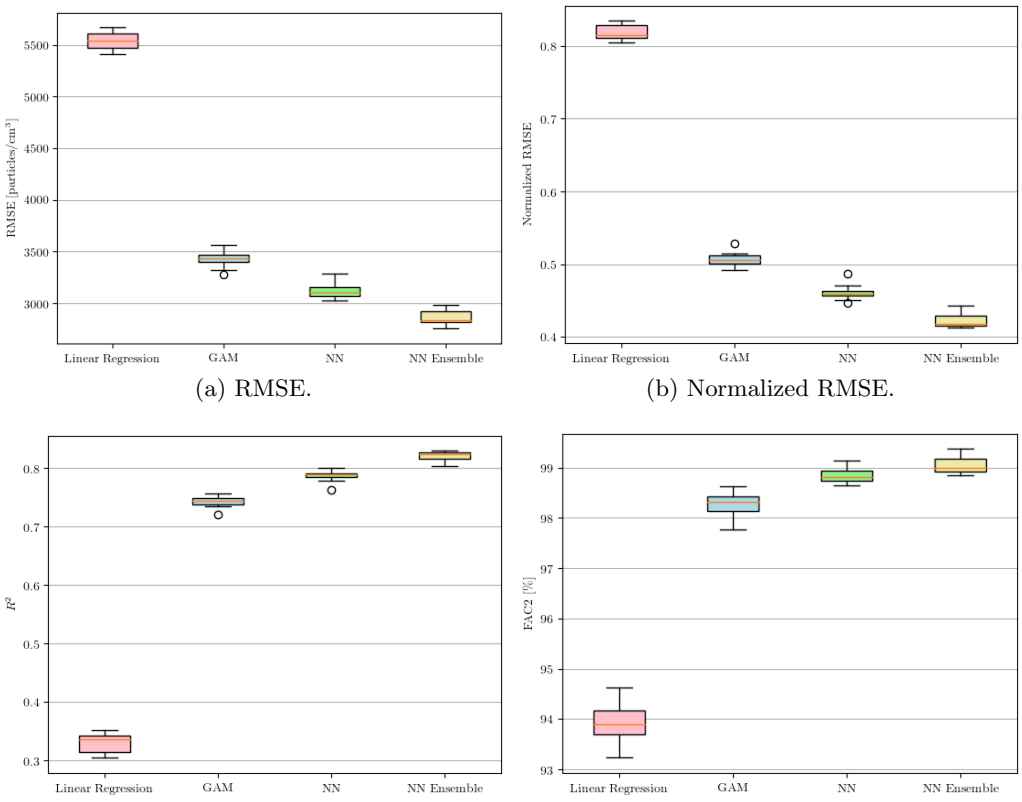
Figure 4.4: Air pollution maps of the month August generated by GAM and NN.



(a) RMSE. (b) Normalized RMSE.

(c) R^2. (d) Factor of 2.

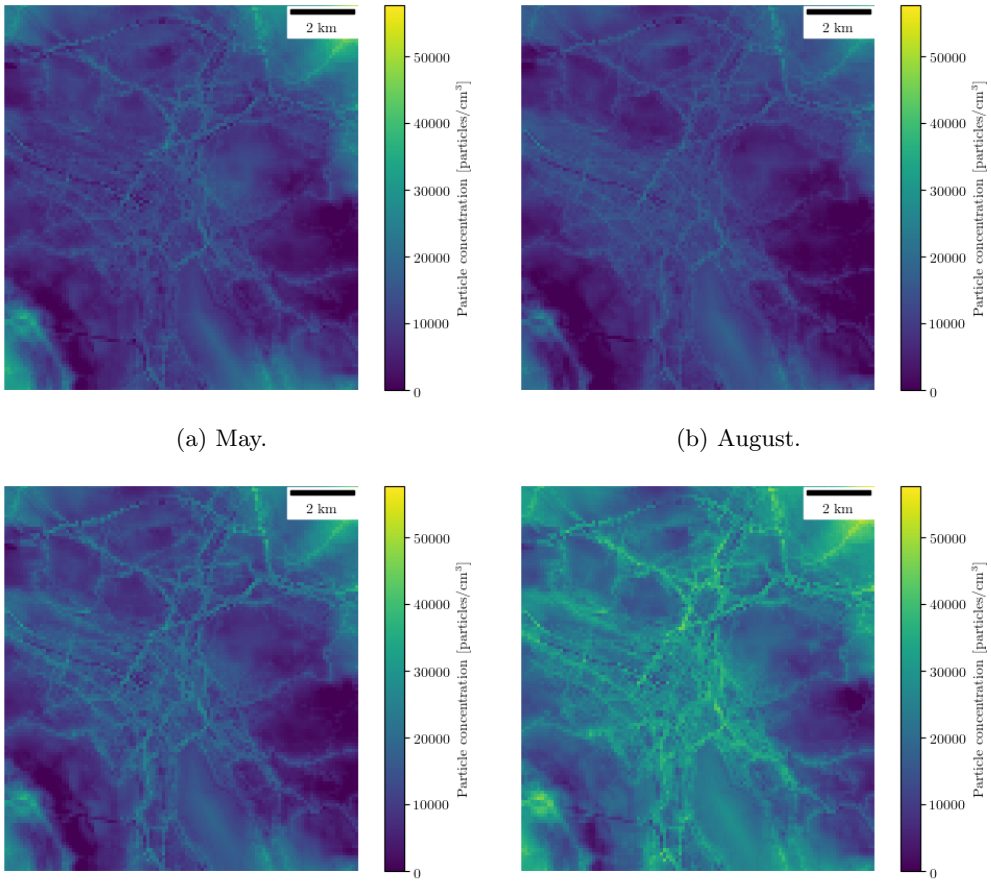
Figure 4.5: Comparison of different structures of the neural network with daily resolution.



(a) RMSE. (b) Normalized RMSE.

(c) R^2. (d) Factor of 2.

Figure 4.6: Comparison of different structures of the neural network with daily resolution.



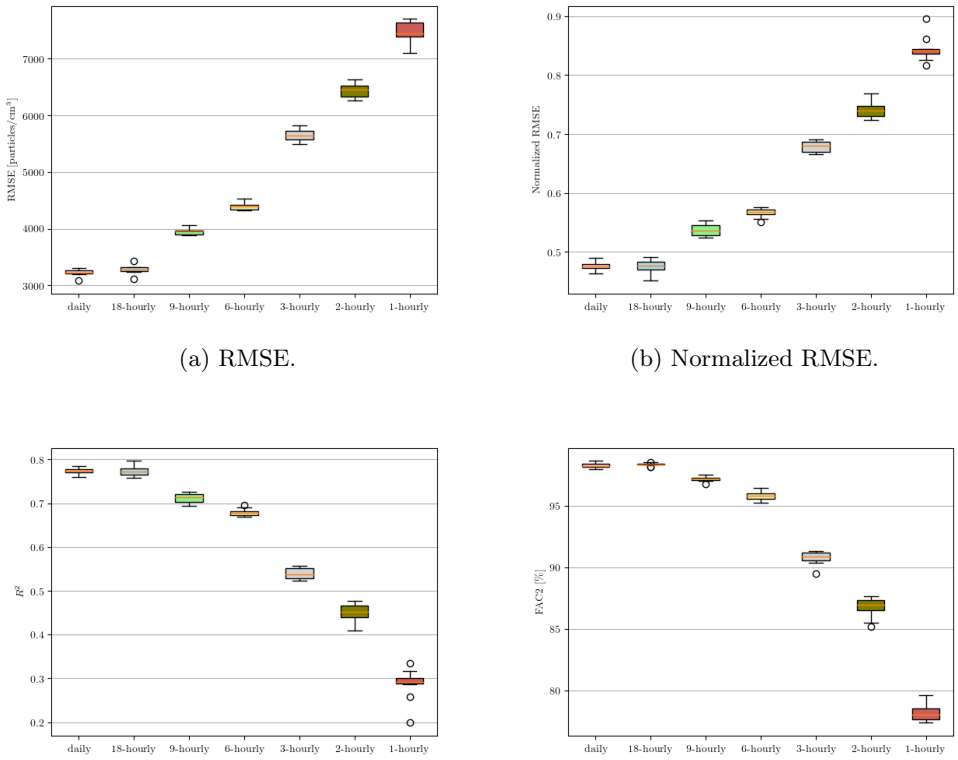
(a) May.

(b) August.

(c) November.

(d) February.

Figure 4.7: Ultra-fine particle maps for one day in May, August, November and February.



(a) RMSE.

(b) Normalized RMSE.

(c) R^2.

(d) Factor of 2.

Figure 4.8: Evaluation of daily to hourly time scales.

4.2.3 Uncertainty estimation

In this subsection, the quality of the uncertainty estimation models is discussed. Furthermore, the spatial and temporal dependencies as well as the feature dependencies of uncertainty maps are evaluated in a sensitivity analysis. For each evaluation, a testset is extracted considering only grid cells with a distance of at least 420 meters from each other.

Quality of uncertainty estimation models

For estimating the quality of the uncertainty of the dropout (Subsection 2.2.2) and ensemble (Subsection 2.2.1) models presented, calibration curves, rank histograms and a bin evaluation are used. The dropout approach is tested using a Monte Carlo estimation with $M = 20$ (RDeepSense-MC20) and using the proposed efficient variation (RDeepSense). For the ensemble approach, 20 neural networks are trained independently and the variance of the mixture of Gaussians (DeepEnsembles-20) and the empirical variance of the outputted means (EmpiricalVariance-20) is taken as an uncertainty measure. For both methods, a neural network with two hidden layers of 50 nodes each is trained on the daily resoluted dataset for 5000 epochs. A dropout rate of 80% is chosen and the tune parameter for the loss function in RDeepSense is chosen such that it consists of 60% mean square error and 40% negative log-likelihood. The geographically spreaded testset is used to produce calibration curves, evaluation in bins and rank histograms (Figure 4.9).
The calibration curves show that both RDeepSense-MC20 and DeepEnsembles-20 are well calibrated, while RDeepSense and EmpiricalVariance-20 tend to underestimate uncertainty predictions. In average, all the models output a higher variance if a higher absolute error is made when predicting the pollution concentration of a sample. The predicted variance of the ensemble method shows the biggest increase while there is only a small increase of the empirical variance. A U-shaped rank histogram is observed for both the ensemble and the dropout model. This could be due to some low and some high biases in the models or the fact that the ensemble does not spread enough. However, the probability distribution of the pollution concentration is not well represented by the predicted means of the ensemble or dropout members.
In Figure 4.10, the generated pollution and uncertainty maps of these approaches are shown, where the predicted variance is thresholded at 1.5. The structure of the uncertainty maps of DeepEnsembles-20 and EmpiricalVariance-20 are similar, but lower variances are predicted by the latter. As expected from the calibration curves, the empirical variance underestimates the uncertainty. Even lower variances are predicted by RDeepSense and a similar structure as in the ensemble maps is observed for the well calibrated RDeepSense-MC20.
Because of the bad performance regarding the error metrics of RDeepSense and

the poor calibration of this approach as well as EmpiricalVariance-20, only the other two models are used for further evaluation.

Sensitivity analysis - spatial dependencies

In order to investigate the effect on the uncertainty of using training datasets with different spatial distributions, three datasets are aggregated which vary in the spatial spread of the samples (Figure 4.11). Two stationary datasets are aggregated considering only grid cells that are at least 300 meters or 2000 meters away from each other. The third dataset contains only grid cells that are at most 1200 meters away from the city center. In Table 4.3, the error metrics of RDeepSense-MC20 and DeepEnsembles evaluated on the geographically spreaded testset are presented. The neural networks trained only on points with 2000 meters distance perform poorly. Moreover, both models also struggle when considering only central datapoints. Unfortunately, the calibration plots show that the uncertainty metrics are also less reliable when using the training sets that result in poor performance (Figure 4.12). However, the bin plots (4.13) show that variance is still increasing with absolute error. As DeepEnsembles-20 achieved the most consistent performance, the uncertainty maps for the different training sets are shown for this approach in Figure 4.14.
At first glance, the network that is only trained on datapoints that are at least two kilometers away from each other should be the best as it shows the biggest region with a low uncertainty. Anyway, the model tends to underestimate uncertainty with this training set as can be seen in the calibration plot.
The network that is only trained on central data shows a higher uncertainty for outer regions in general. Furthermore, there are some lines (streets) where the model predicts a high uncertainty.

		Training datasets			
		All	**300m**	**2000m**	**central**
RDeepSense-MC20	RMSE	4370.59	4731.21	9781.83	5911.89
	NRMSE	0.62	0.675	1.396	0.844
	R^2	0.61	0.544	-0.95	0.288
	FAC2	96.16	95.69	76.29	90.61
DeepEnsembles-20	RMSE	4061.58	4278.85	5694.48	4821.51
	NRMSE	0.58	0.611	0.813	0.688
	R^2	0.66	0.627	0.339	0.526
	FAC2	96.82	96.61	92.77	95.49

Table 4.3: Error metrics of the dropout and ensemble approach with spatially different training sets.

Sensitivity analysis - feature dependencies

In order to investigate the effect of different features on the uncertainty, features that highly correlate to the pollution concentration are removed. The following features are removed from the dataset:

a) distance to traffic features

b) wind velocity, wind gust and wind strength

c) all traffic-related features

Performance only decreased significantly when removing all traffic-related features (Table 4.4). For all the different feature sets, DeepEnsemble-20 calibrates well. The generated air pollution concentration and the uncertainty maps that go with them are shown in Figure 4.15.
Removing the wind features does not influence the uncertainty maps much as these aggregated features do not differ spatially. However, removing spatially resoluted features like traffic can lead to a remarkable change in both pollution and uncertainty maps. When the 'distance of traffic' features are removed, air pollution maps do not change much. However, at some regions, the variance is lowered significantly compared to the uncertainty maps generated using all features. Apparently, even if some features do not have a big impact on the predicted pollution concentration, they can lead to a very high uncertainty at locations where the feature values differ from the training values considerably.

		Feature sets		
		a)	**b)**	**c)**
DeepEnsembles-20	RMSE	4042.06	4096.67	4631.889
	NRMSE	0.577	0.585	0.661
	R^2	0.667	0.658	0.563
	FAC2	96.91	96.736	95.304

Table 4.4: Error metrics of the ensemble approach with different features used for training.

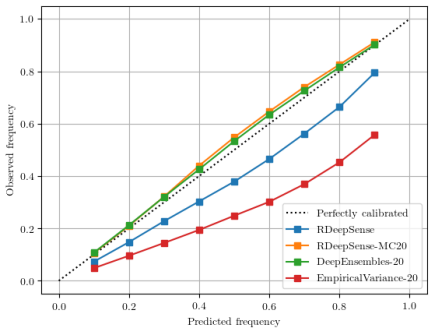
Sensitivity analysis - temporal dependencies

In this part, the temporal dependencies of the uncertainty of air pollution inference is investigated. Here, the models are trained and tested on the data of only one month. In most cases, NRMSE and R^2 of training on a monthly basis are worse than if the dataset of one whole year is used for training (Table 4.5). However for November 2012, a lower NRMSE and higher R^2 is achieved, and at

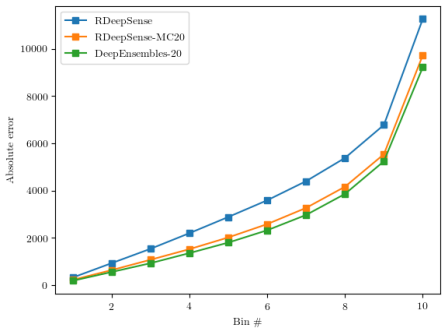
		Training datasets			
		May 12	**August 12**	**November 12**	**February 13**
DeepEnsembles-20	RMSE	3067.543	3290.389	3857.422	5547.23
	NRMSE	0.695	0.601	0.544	0.601
	R^2	0.516	0.639	0.704	0.638
	FAC2	97.698	98.162	96.226	95.918

Table 4.5: Error metrics of the ensemble approach trained on different months.

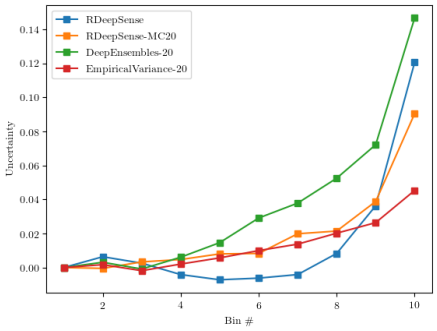
the same time, the lowest uncertainty is reached for this month (Figure 4.16). The model seems to yield better and more certain air pollution predictions for the month of November compared to the other months.
In order to investigate the uncertainty development over three years, some weather features (wind gust, wind velocity, wind chill, air pressure, dew point and rain) and the features extracted from the vehicle counting cells are left out because these features are only extracted from April 2012 to March 2013. The daily air pollution and uncertainty maps generated for one day in May 2012, 2013 and 2014 are shown in Figure 4.17. There is a high temporal dependency in uncertainty estimations. However, no clear tendency is observed that air pollution predictions get more uncertain when using pollution data from older sensors.



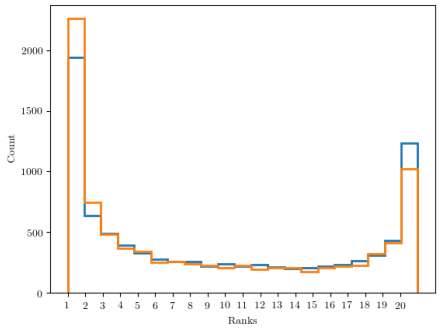
(a) Calibration curves.



(b) Mean of absolute error in bins.

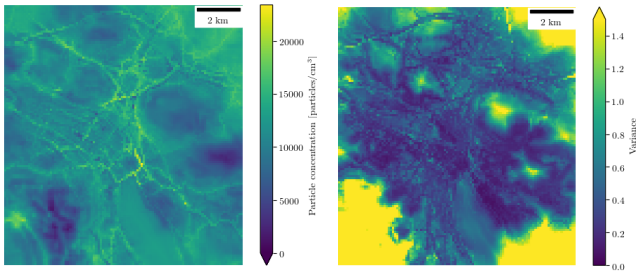


(c) Mean of variance in bins.

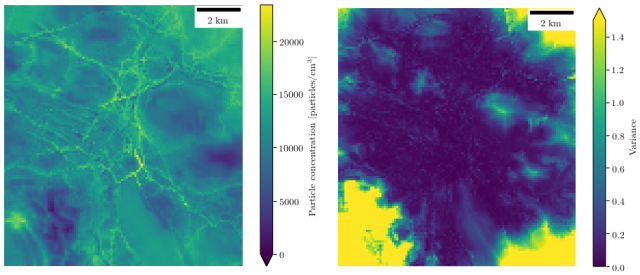


(d) Rank histogram.

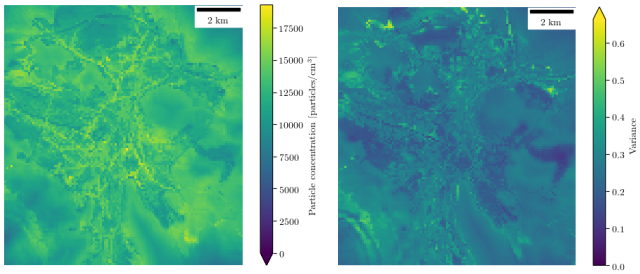
Figure 4.9: Evaluation of uncertainty estimations for the dropout and ensemble models.



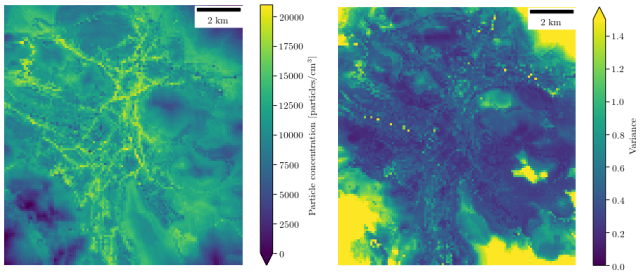
(a) DeepEnsembles-20.



(b) EmpiricalVariance-20.

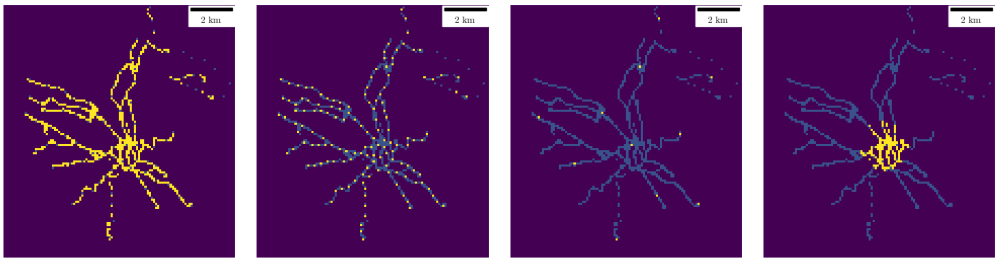


(c) RDeepSense.



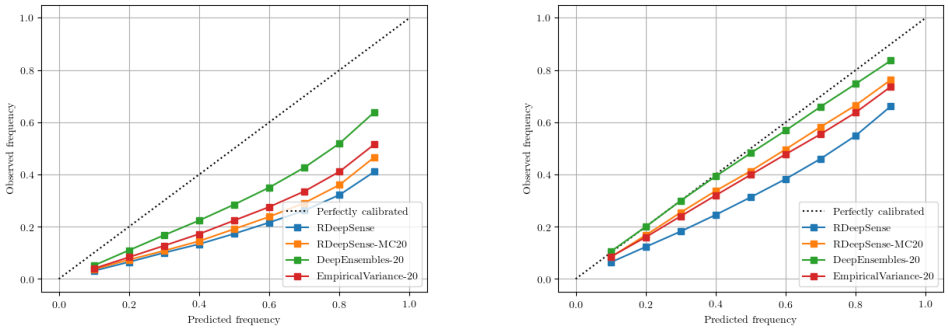
(d) RDeepSense-MC20.

Figure 4.10: Air pollution (left) and uncertainty maps (right) of ensemble and dropout models.



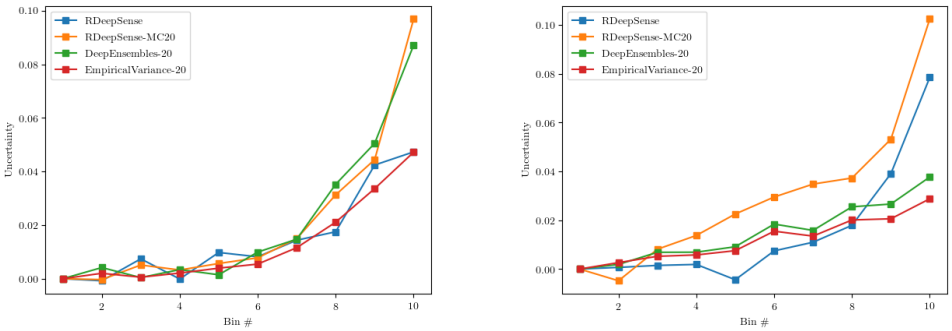
(a) All. (b) Stationary (300m). (c) Stationary (2000m). (d) Central data.

Figure 4.11: Overview of the datasets used for training (yellow).



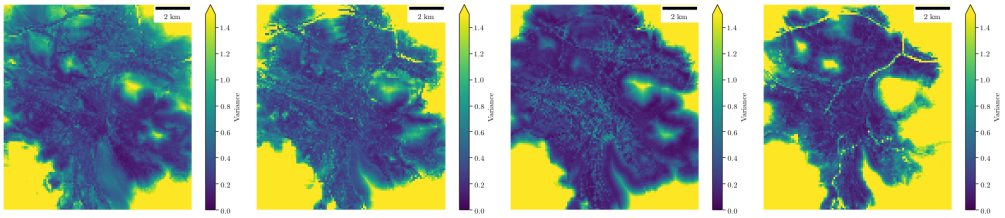
(a) Stationary (2000m). (b) Central data.

Figure 4.12: Calibration curves using the spatially different training datasets.



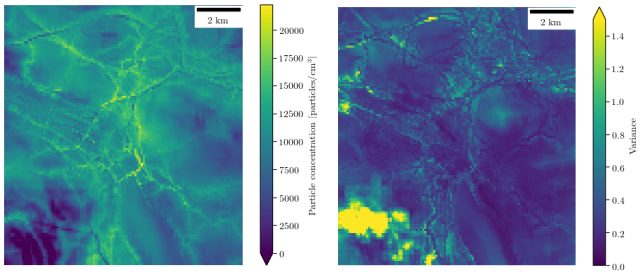
(a) Stationary (2000m). (b) Central data.

Figure 4.13: Bin evaluation using the spatially different training datasets.

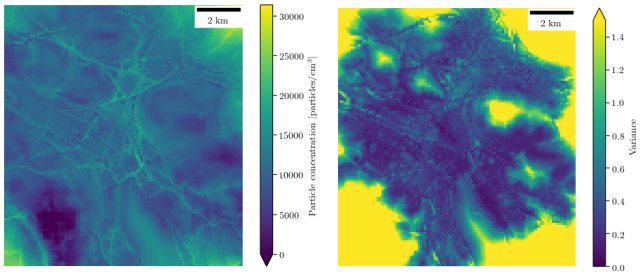


(a) All. (b) Stationary (300m). (c) Stationary (2000m). (d) Central data.

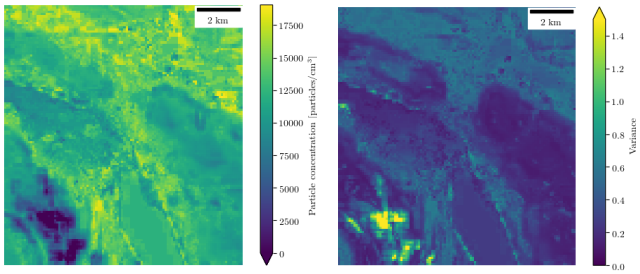
Figure 4.14: Uncertainty maps generated by DeepEnsembles-20 for spatially different training datasets.



(a) Remove distance of traffic features.

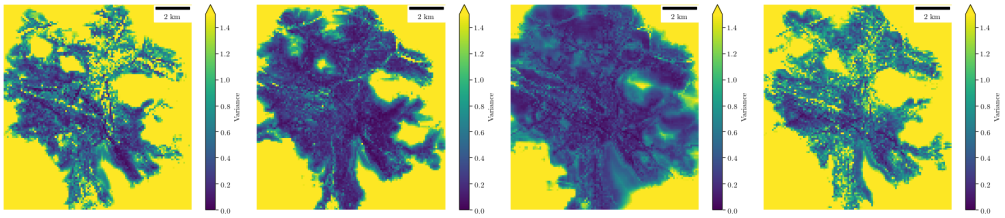


(b) Remove wind velocity, wind gust and wind strength.



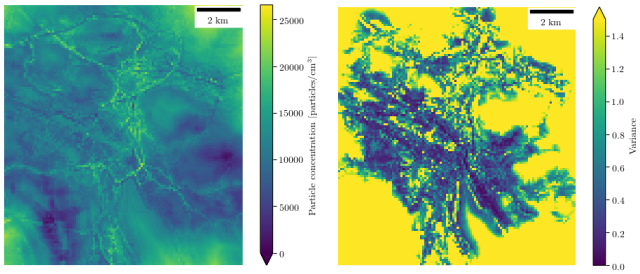
(c) Remove all traffic-related features.

Figure 4.15: Comparison of air pollution maps (left) and uncertainty maps (right) of DeepEnsembles-20 using different sets of features.

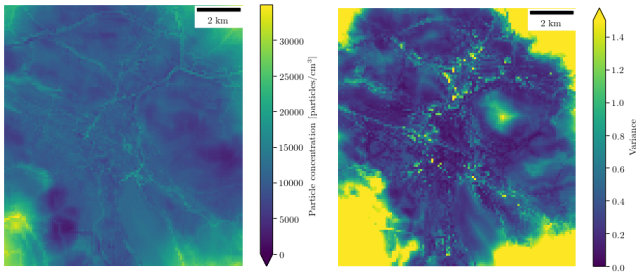


(a) May 2012. (b) August 2012. (c) November 2012. (d) February 2013.

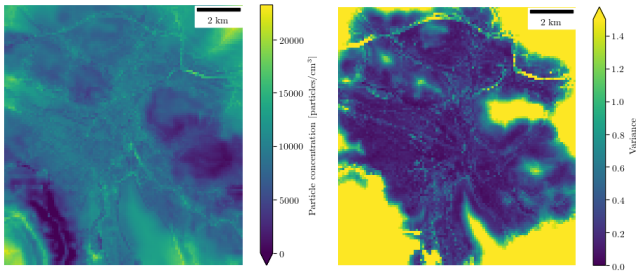
Figure 4.16: Comparison of uncertainty maps of DeepEnsembles-20 trained on different months.



(a) May 2012.



(b) May 2013.



(c) May 2014.

Figure 4.17: Comparison of air pollution maps (left) and uncertainty maps (right) of DeepEnsembles-20 trained on the daily data of one month.

Chapter 5

Conclusion

5.1 Discussion

In this thesis, an air pollution map generation method based on neural networks is presented. Air quality in the city of Zurich is inferred using ultra fine particle measurements recorded by a mobile sensor network and three types of features (land use, weather and traffic). The dependencies between features and air pollution is learned by training a neural network on the measurements of the sensor network. At locations and times where no measurements are available, the trained network is used to extrapolate the air pollution concentration for the whole area of Zurich. Additionally, approaches that provide an uncertainty measure of the estimated air pollution concentration are evaluated and uncertainty maps are generated.

The comparison with the land use regression model showed that a neural network is a good alternative for modelling the dependencies of land use features and air pollution concentration. Even a simple neural network reached a similar performance as the land-use regression model. In order to increase the temporal resolution, timely resolved weather and traffic features are added to the land use features and a daily resolved dataset including air pollution and feature data from one year is aggregated. An ensemble of neural networks outperformed single neural networks as well as a generalized additive model and a linear regression model. Weather features showed to be important for estimating air pollution, whereas traffic features showed to be less important than land use and weather features. A good inference of ultra fine particles could be reached up to 6-hourly resolution. However, for higher temporal resolutions, the model performance drops to an unsatisfactory level.
Even though the neural network model showed improved error metrics compared to GAM, the inspection of the generated air pollution maps uncovers the disadvantage of this model. Unrealistic pollution concentrations are predicted at locations where the input features differ significantly from the feature values of the training dataset. The error metrics are not able to capture this restriction because the test dataset contains only points on the tram network.

A well calibrated uncertainty estimation is provided by the ensemble and dropout approaches. The U-shaped rank histograms showed that the probability distribution of the pollution concentration is not well represented by the ensemble and dropout members. However, a tendency that estimated variance increases with absolute error is observed.
The generated uncertainty maps show that neural networks are uncertain in its air pollution predictions at locations that have a high distance to the training data. More precisely, the uncertainty is high at grid cells where a feature value differs a lot from the feature values that are feeded into the neural network. This behaviour was already observed by the generated air pollution maps and is confirmed by the uncertainty maps. Uncertainty predictions are highly sensitive regarding the removal of spatially resolved features. Removing such features may lower the predicted uncertainty by the network, but will not make the inferred air pollution concentration better, as the effects of these features on the air pollution concentration are not considered anymore.

To sum up, the neural network approach performed good for estimating air pollution at points with similar environmental conditions like the original dataset. The temporal resolution is captured well by the neural network approach because the full range of timely resolved weather and traffic values are contained in the training set. However, this approach is not able to accurately extrapolate air pollution concentrations at other locations because the features are too divergent. In this case of having air pollution measurements from a mobile tram network, a neural network is only promising for high quality air pollution predictions at locations similar to the samples in the training data. However, the ability of extrapolating air pollution concentrations at other locations is doubted. The generalized additive model from the land use regression model showed more promising abilities for extrapolation, but no information about the uncertainty of the predictions is provided there.

5.2 Future work

An improvement of the temporal resolution of proposed models could be reached by aggregating more timely resolved features like human mobility or traffic features that include the speed of vehicles [8]. In order to improve the spatial resolution, the inputs to the neural network have to be in a similar range at all locations of the map. Therefore, a preprocessing of the features is inevitable. One approach could be to compute correlations between the features and use them as input to the neural network [8]. A co-training approach of using neural networks for timely resolved features and a GAM for spatially resolved features could reach a better performance for extrapolating UFP concentrations.

Only a tendency of correlation between predicted variance and absolute error is observed by the uncertainty estimation approaches. A low predicted variance

tends to result in a low absolute error, but a high error can be present for individual samples. This limits the usage of these uncertainty methods, as the measures are not trustworthy for single predictions. However, both approaches could be used to detect out-of-distribution examples when using a neural network model.

Appendix A

Reconstruction of the results

A.1 Dataset aggregation

New weather data and traffic data

- In `data/weather_data/`, the weather data is stored in 10 minutes and daily resolution. `weather_daily.npy` and `weather_hourly.npy` are generated by `python weather_hourly.py` and `python weather_daily.py` and contain the daily or hourly averages.
- In `data/traffic_data/Gesamtverkehrsmodelle/gvm.py`, the DTV and DWV are in stored `.mat` format.
- The data of the vehicle counting cells is prepared in `data/traffic_counts.py`

Aggregation of daily to hourly datasets

For the evaluation of daily to hourly temporal resolution, the features are allocated to the pollution data of one year by `dataset_aggregation/add_features_daily.py` and `dataset_aggreagation/add_features_hourly.py`.

In Table A.1 and A.2, the structure of the daily (`daily_50_allfeatures2.npy`) and hourly (`Nhourly_allfeatures50.npy`) datasets are shown. Here, x and y are the coordinates of the grid cell, c_i is the pollution concentration and features are the allocated land use, traffic and weather features. The order of the allocated features is shown in Table A.3.

day	x	y	c_i	features

Table A.1: Columns of daily dataset

day	hour	x	y	c_i	features

Table A.2: Columns of N-hourly dataset

0	population	16 - 21	dtv 1-6
1	industry	22	len_streets
2	floorlevel	23 - 28	dwv 1-6
3	heating	29	temperature
4	elevation	30	humidity
5	streetsize	31	wind gust
6	signal_dist	32	wind velocity
7	street_dist	33	wind strength
8	slope	34	wind direction
9	exp_slope	35	wind chill
10	traffic	36	air pressure
11	streetdist_m	37	dew point
12	streetdist_l	38	rain
13	trafficdist_l	39	global radiance
14	trafficdist_h	40	cars_1
15	traffic_tot	41	cars_2

Table A.3: Order of the features in the dataset

A.2 LSTM results

Forecasting based on weather features

The implementation of this experiment is based on this example. and is located in `lstm_nn/`. By running `python lstm.py`, the error metrics are printed.

Comparison of LSTM with the fully-connected neural network

The code for the comparison of the LSTM with the fully-connected neural network is also stored in `lstm_nn/`.

1. adjust parameters in `config2.yaml`: set `model` and `job` to `'both'` for a comparison of NN and LSTM.
2. run `python daily.py`, a directory including the results is created in `results/`

3. outputs: `summary.csv` (error metrics of LSTM and NN), `y_lstm_nn0.png` (plot of absolute error of NN vs absolute error of LSTM)

A.3 NN results

The experiments for yearly to biweekly resolution are stored in `nn_infer1/` and the code for daily to hourly resolution can be found in `nn_infer2/`.

Yearly to biweekly resolution

1. adjust parameters in `config.yaml`, choose data (yearly, seasonal, monthly or biweekly) in `datadir` (air pollution data with features)
2. for generating maps:
   (a) set `job` to `'maps'`
   (b) run `python run.py`
   (c) plot outputted maps (`nn_model_pm_ha_ext_***.mat`) with matlab function `plot_model_map.m`

   for CV tests:
   (a) set `job` to `'CV_test'`
   (b) run `python run.py`
   (c) for plotting CV results: `results_nngam.py`

Daily to hourly resolution

For training individual models:

1. adjust parameters in `config_nn.yaml`:
   (a) `dataset`: `../data/daily_50_allfeatures2.npy` for daily resolution, `../data/*hourly_allfeatures50.npy` for hourly resolutions
   (b) `layers`: choose structure of nn models
   (c) `model`: `'nn'` or `'gam'`
   (d) `job`: `'onetest'` or `'CV_test'`
   (e) `remove_features`: set to `True` and specify indices in `features` if features should be removed
2. run `python train_nn.py`

3. plot CV results with `results_daily.py` or `results_hourly.py`

For combining the models to an ensemble:

1. adjust parameters in `config_ensemble.yaml`:
   (a) `models_dir`: path where the trained models are stored
   (b) `job`: `'onetest'` or `'CV_test'`, choose `'onetest'` for generating pollution maps
2. run `python ensemble.py`

A.4 Uncertainty results

The implementation of the uncertainty methods is based on this implementation and is located in `'nn_uncertainty/'`.
To train and test both ensemble and dropout methods, execute `python train.py --args ARGVALUE` or `python test.py --args ARGVALUE`, and adjust desired arguments (Table A.4).

Remarks:

- for feature evaluation (`preproc_mode` 4), make sure to adapt the first entry of `args.sizes` in `main()` and the specify features that should be removed in `utils.py`
- for monthly training and testing (`preproc_mode` 5), make sure to adapt month and year in `utils.py`.
- trained models are stored in `nn_uncertainty/euler_results/sensitivity_analysis/`
- for testing, specify path of trained models in `main()` and make sure to use the same `args` as trained models
- change from `DataLoader_AirPollutionDaily()` to `DataLoader_AirPollutionDaily_temporal()` for monthly evaluation after April 2013

`args`	`ARGVALUE`
`ensemble_size`	Size of ensemble (default: 20)
`epochs`	Number of epochs for training (default: 5000)
`batch_size`	Size of batch (default: 50)
`epsilon`	Epsilon for adversarial input perturbation for ensemble method (default: 0.02)
`alpha`	Trade off parameter for likelihood score and adversarial training for ensemble method (default: 0.5)
`loss_alpha`	Trade off parameter for MSE and NLL loss of dropout approach (default: 0.6)
`keep_prob`	Keep probability for dropout (default: 0.8)
`preproc_mode`	Specify preprocessing of dataset (default: 0) 0: no preprocessing 1: PCA 2: stationary dataset 3: central dataset 4: remove features 5: train/test on month
`distance`	Distance or range of training samples (default: 2000) preproc_mode 2: specifies distance between training samples preproc_mode 3: specifies range from center of training samples

Table A.4: Arguments for `train.py` and `test.py`

Bibliography

[1] (2018) Opensense webpage. [Online]. Available: https://gitlab.ethz.ch/tec/public/opensense

[2] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, T. Arn, J. Beutel, and L. Thiele, "Deriving high-resolution urban air pollution maps using mobile sensor nodes," vol. 16, 12 2014.

[3] M. Mueller, D. Hasenfratz, O. Saukh, M. Fierz, and C. Hueglin, "Statistical modelling of particle number concentration in zurich at high spatio-temporal resolution utilizing data from a mobile sensor network," vol. 126, 12 2015.

[4] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, "Forecasting fine-grained air quality based on big data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 2267–2276. [Online]. Available: http://doi.acm.org/10.1145/2783258.2788573

[5] J. Hooyberghs, C. Mensink, G. Dumont, F. Fierens, and O. Brasseur, "A neural network forecast for daily average pm10 concentrations in belgium," *Atmospheric Environment*, vol. 39, no. 18, pp. 3279 – 3289, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1352231005001408

[6] H. Niska, T. Hiltunen, A. Karppinen, J. Ruuskanen, and M. Kolehmainen, "Evolving the neural network model for forecasting air pollution time series," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 2, pp. 159 – 167, 2004, intelligent Control and Signal Processing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197604000119

[7] M. A. Elangasinghe, N. Singhal, K. N. Dirks, and J. A. Salmond, "Development of an ann–based air pollution forecasting system with explicit knowledge through sensitivity analysis," *Atmospheric Pollution Research*, vol. 5, no. 4, pp. 696 – 708, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1309104215302786

[8] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: When urban air quality inference meets big data," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser.

KDD '13. New York, NY, USA: ACM, 2013, pp. 1436–1444. [Online]. Available: http://doi.acm.org/10.1145/2487575.2488188

[9] A. Marjovi, A. Arfire, and A. Martinoli, "Extending urban air quality maps beyond the coverage of a mobile sensor network: Data sources, methods, and performance evaluation," 2 2017.

[10] H.-P. Hsieh, S.-D. Lin, and Y. Zheng, "Inferring air quality for station location recommendation based on urban big data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 437–446. [Online]. Available: http://doi.acm.org/10.1145/2783258.2783344

[11] L. Chen, Y. Cai, Y. Ding, M. Lv, C. Yuan, and G. Chen, "Spatially fine-grained urban air quality estimation using ensemble semi-supervised learning and pruning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16. New York, NY, USA: ACM, 2016, pp. 1076–1087. [Online]. Available: http://doi.acm.org/10.1145/2971648.2971725

[12] A. Jutzeler, J. J. Li, B. Faltings *et al.*, "A region-based model for estimating urban air pollution." in *AAAI*, 2014, pp. 424–430.

[13] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6402–6413. [Online]. Available: http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf

[14] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 1, June 1994, pp. 55–60 vol.1.

[15] S. Yao, Y. Zhao, H. Shao, A. Zhang, C. Zhang, S. Li, and T. Abdelzaher, "Rdeepsense: Reliable deep mobile computing models with uncertainty estimations," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 173:1–173:26, Jan. 2018. [Online]. Available: http://doi.acm.org/10.1145/3161181

[16] N. Li, C. Sioutas, A. Cho, D. Schmitz, C. Misra, J. Sempf, M. Wang, T. Oberley, J. Froines, and A. Nel, "Ultrafine particulate pollutants induce oxidative stress and mitochondrial damage." *Environmental Health Perspectives*, vol. 111, pp. 455–460, 2013.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[19] D. A. Nix and A. S. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 1, June 1994, pp. 55–60 vol.1.

[20] S. Pfreundschuh, P. Eriksson, D. Duncan, B. Rydberg, N. Håkansson, and A. Thoss, "A neural network approach to estimating a posteriori distributions of bayesian retrieval problems," *Atmospheric Measurement Techniques*, vol. 11, no. 8, pp. 4627–4643, 2018. [Online]. Available: https://www.atmos-meas-tech.net/11/4627/2018/