Neural Kernel Network Deep Kernel Learning for Predicting Particulate Matter from Heterogeneous Sensors with Uncertainty

Chaofan Li(✉), Till Riedel, and Michael Beigl

Karlsruhe Institute of Technology, Vincenz-Prießnitz-Str. 1,
76131 Karlsruhe, Germany
chaofan.li@kit.edu

Abstract. Modern web technologies allow novel types of sensor networks that collect measurements from different sources ranging from citizen-collected data to official sources. In this paper, we propose a scheme to deal with measurement sources of different quality for time-series prediction of urban particulate matter. Our approach is based on a neural kernel network deep kernel learning model that takes the highly heterogeneous and uncertain measurements provided by a post-hoc hybrid low-cost sensor network as input. And predicts the daily average PM10 mass concentration for the next day. Furthermore, we also validate the contribution of ultra-low-cost sensors in the SmartAQnet sensor network, which reduces the average MAE of our model pipeline from $4.18\,\mu g/m^3$ to $3.67\,\mu g/m^3$ and increases the PCC from 0.589 to 0.665. In contrast to existing approaches, we can model the uncertainty of the prediction based on the quality and quantity of input signals crawled.

Keywords: Machine learning · Gaussian process regression · Time-series prediction · Urban air quality · Low-cost sensor network

1 Introduction

Air pollution causes severe damage to human health. According to the WHO Air Quality Guidelines 2021 [10], outdoor and household air pollution accounted for about 12% of all deaths in 2019. Moreover, even exposure to low concentrations of air pollutants is dangerous, and no thresholds that could be considered safe were observed. Among various air pollutants, Particulate Matter (PM) is causally associated with all-cause mortality and diseases such as acute lower respiratory infections, lung cancer, etc. [6] Individual exposition can vary extremely within the same city. Therefore, fine-grained monitoring and forecasting of PM concentrations in urban areas have become an important task.

In urban air quality modeling, statistical modeling approaches based on machine learning (ML) algorithms are receiving more and more attention as they do not rely on accurate data on emission sources in rapidly changing urban areas. ML algorithms, however, generally require large-scale datasets to capture the relationship between measurements and predictions.

In practical applications, a trade-off must be made between the network size of a sensor network and the average cost of sensors due to budget constraints. Rather than using a dedicated measurement network, post-hoc networks combine the small number of high-precision measurement stations for which government agencies publish data on the internet with low-cost sensors run collaboratively by citizens and researchers [2,9]. However, fusing such heterogeneous information into a large-scale urban air quality measurement network is often characterized by high uncertainty.

This paper proposes a new model pipeline based on a neural kernel network [21] deep kernel learning model. It takes heterogeneous and uncertain data collected from different Internet-connected sources by the "SmartAQnet" [2] as the input. And it predicts the daily average of PM10 concentration readings of the four high-precision PM10 monitoring stations for the next day. As a result, our model pipeline achieved an average mean absolute error (MAE) of 3.67 $\mu g/m^3$ and an average Pearson correlation coefficient (PCC) of 0.665. We also test the effect of different preprocessing strategies and compare our prediction model with other comparison models (baseline, MLP, LSTM, vanilla GPR, etc.). Furthermore, we also validate the contribution of ultra-low-cost sensors in the SmartAQnet sensor network, which reduces the average MAE of our model pipeline from 4.18 $\mu g/m^3$ to 3.67 $\mu g/m^3$ and increases the PCC from 0.589 to 0.665.

2 Background

2.1 Urban Air Quality Data

Historical data with reliable reference measurements is an important asset required for modeling tasks. It provides the ground truth necessary to fit and validate models. Traditional urban air quality data is usually collected by sparsely distributed high-precision measurement stations [5]. These measurement stations provide high-quality measurement data but are expensive to install and maintain. This is why typically only a minimal number of such stations are deployed: typically to comply with regulatory requirements. However, urban areas are highly complex environments, so fine-grained monitoring and modeling of urban pollutants require high temporal and spatial resolution data. To this end, many grassroots, scientific, or even in some cases municipality-driven efforts have been made to simultaneously deploy at much lower price levels to balance the spatial density and cost of deployment [2,8,9,15]. Particularly opportunistically combining the established sensor networks of different local agencies is also a potential way to increase the data scale and resolution [2]. However, for those internet of thing (IoT) enabled methods introduce new difficulties in the data, namely the problem of calibration, heterogeneity and uncertainty.

Heterogeneity in IoT Air Quality Sensor Networks. For data from hybrid sensor networks containing sensors at multiple price levels, the problem of heterogeneity may arise from numerous aspects, such as:

Construction and Maintenance of the Sensor Network. The operational stability of a sensor is often positively related to its cost. When the network contains a large number of medium and low-cost sensors, it becomes almost impossible to maintain the stability of the network. Figure 1 shows the average daily available devices for each month during the operation of the SmartAQnet project [2]. In such a network, both the deployment and the failure of sensors are present throughout the time. Some damaged sensors may eventually be repaired or replenished to work at the former location. However, there are also possibilities that the damaged sensor may never come back to operate, especially when it is a low-cost one.

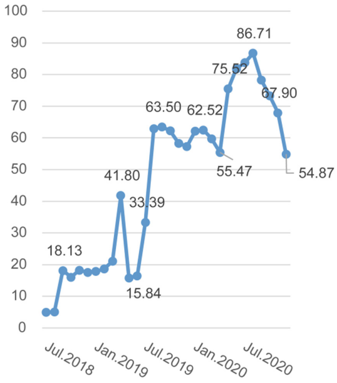


Fig. 1. Average daily available devices for each month

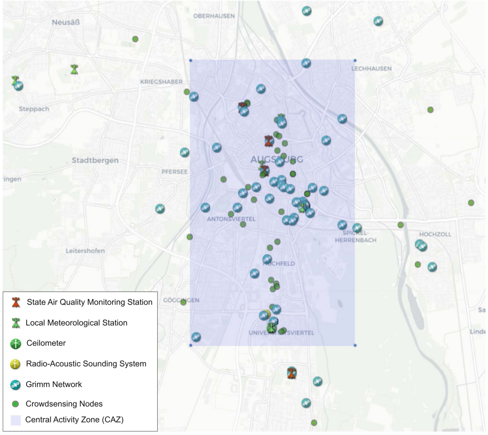


Fig. 2. Sensor deployment map of the SmartAQnet project

Combination of Different Local Sensor Network. Different institutions have different interests in the observed properties, leading them to choose different types of sensors and adopt different spatial and temporal strategies when deploying their own sensor networks. Such differences in strategy will result in differences in the structure of the data sheets, which means there will be a large number of missing values in the merged dataset.

Uncertainty in IoT Air Quality Sensor Networks. Uncertainty issues also come from many different sources in hybrid sensor networks, such as:

Errors of Low-Cost Sensors. The error of low-cost sensors is an essential factor in the uncertainty of the data. For example, Budde et al. analyzed the performance of a popular low-cost PM sensor, SDS011, in [4]. They find out in their experiments that a notable variance between individual sensors can be observed, and the reliability of the sensor's readings is also affected by various environmental factors like humidity. Previous research on citizen weather stations employing also Gaussian process regression, however, to provide spatial interpolation shows the importance but also the complexity of modelling sensor error as uncertainty [1].

Errors of Human Management. The longer the urban air quality sensor network runs, the more likely it is to collect more data across different patterns to help with better modeling. Therefore, such projects tend to last for years and generate hundreds of millions of observational records, posing enormous challenges to project management. In addition, some existing urban air quality projects, such as SmartAQnet [2], Smart Emission [9], and Luftdaten.info, also include citizen science programs, which make human management errors more difficult to detect [3]. No one can guarantee that all human errors are detected and fixed. The associated uncertainty probably still remains in the aggregated data.

2.2 Urban Air Pollutants Modeling

Current urban air pollutant models can be roughly divided into simulation and statistical models. Among them, the simulation model usually makes predictions by simulating the physical and chemical processes of pollutant diffusion and reaction in the atmosphere [16,19]. In comparison, the statistical model makes predictions by summarizing the statistical characteristics from historical observations [13,17,18,20]. Our model pipeline adopts a Gaussian process-based statistical modeling approach.

Gaussian Process Regression. Gaussian Process Regression (GPR) is a classic non-parametric Bayesian regression algorithm. It has the advantage of performing well on small datasets and provides a measure of uncertainty for predictions.

Unlike many popular supervised machine learning algorithms such as MLP and LSTM, Bayesian methods don't just learn an exact value for each parameter in a function. They infer the probability distribution of the parameter over all possible values. The way Bayesian methods work is to specify a prior distribution $p(w)$ for the parameter w and then use the Bayesian rule (Eq. 1) to relocate this probability distribution based on the evidence (that is, observational data).

$$p(w|y,X) = \frac{p(y|X,w) \times P(w)}{p(y|X)} \tag{1}$$

The relocated probability distribution $p(w|y,X)$ is called the posterior distribution, containing information from both the prior distribution and the dataset. When we want to predict the label of a point of interest x^*, the predictive distribution can be calculated by weighting all possible predictions by their posterior distribution (Eq. 2).

$$p(f^*|x^*, y, X) = \int_w p(f^*|x^*, w)p(w|y, X)dw \tag{2}$$

Instead of calculating a probability distribution over the parameters of a particular function, GPR calculates a probability distribution over all possible functions that fit the data. In GPR, we first assume a Gaussian process prior, which can be defined by a mean function $m(x)$ and a covariance function $k(x, x')$. The training set and the predicted points of interest are joint multivariate Gaussian distributed from the Gaussian process prior (Eq. 3). The training process of GPR is to find suitable parameters for the mean and covariance functions. This is usually done with the help of a gradient-based optimizer by maximizing the log marginal likelihood on the training set.

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu^* \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix}\right) \tag{3}$$

3 Methodology

3.1 Data Description

The data used in this paper is freely available on the internet and aggregated by the www.smartaq.net website. For reproducibility, we are using a dataset provided by the SmartAQnet project that covers all measurements from January 1, 2017, to December 31, 2021 [14].

SmartAQnet combines meteorology and aerosol measurement data collected by different entities. A considerable portion of the sensors is located in a rectangular area of 6 × 4 km that covers most of the city of Augsburg, Germany (Fig. 2). The time resolution of the sensor varies by its model. Among them, the high-precision PM10 measuring station generates a record every 1 h, and the temporal resolution of the vast majority of the other sensors is between 5 min and 5 s.

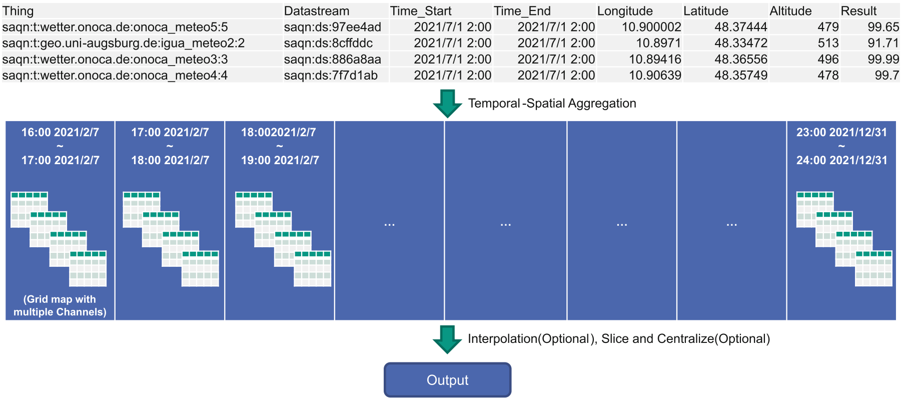
In this study, we treat the data from the above-mentioned 4 high-precision PM10 measurement stations as labels. Other data are treated as input data. Several sensors with remote locations and all height profile data are removed. The considered sensors are all located in a rectangular area of about 16 × 16 km. Among over 30 observed properties provided in the dataset, we select 9 properties that we believe are highly correlated with PM10 concentration to be input into our model. Namely PM10 mass concentration, PM2.5 mass concentration, temperature, relative humidity, air pressure, precipitation, wind direction, wind speed, and global radiation.

3.2 Model Pipeline

Our model pipeline could be divided into three steps: data preprocessing, feature extraction, and prediction.

Data Preprocessing. The data preprocessing step is responsible for receiving the readings directly from the sensor network and preliminarily eliminating heterogeneity and uncertainty in the input data through methods such as aggregation and interpolation. The data preprocessing step can be further divided into two stages: temporal-spatial aggregation and window generation.

Temporal-Spatial Aggregation. Data aggregation is a simple and effective way to reduce data uncertainty, especially good at dealing with the influence of symmetrically distributed noise and small probability events. Thanks to the high temporal and spatial resolution of the SmartAQnet network, we can aggregate a considerable number of observation records into one. In this stage, the aggregation will be carried out on all input data on the time and space dimensions (Fig. 3). All data from the four high-precision PM10 measurement stations, which will be used as labels, are excluded from data aggregation.



Thing	Datastream	Time_Start	Time_End	Longitude	Latitude	Altitude	Result
saqn:t:wetter.onoca.de:onoca_meteo5:5	saqn:ds:97ee4ad	2021/7/1 2:00	2021/7/1 2:00	10.900002	48.37444	479	99.65
saqn:t:geo.uni-augsburg.de:igua_meteo2:2	saqn:ds:8cffddc	2021/7/1 2:00	2021/7/1 2:00	10.8971	48.33472	513	91.71
saqn:t:wetter.onoca.de:onoca_meteo3:3	saqn:ds:886a8aa	2021/7/1 2:00	2021/7/1 2:00	10.89416	48.36556	496	99.99
saqn:t:wetter.onoca.de:onoca_meteo4:4	saqn:ds:7f7d1ab	2021/7/1 2:00	2021/7/1 2:00	10.90639	48.35749	478	99.7

Fig. 3. The workflow of the Temporal-Spatial Aggregation stage

In the time dimension, the aggregation resolution is 1 h, while in the spatial dimension, the spatial extent covered by the dataset (about 16 × 16 km, as mentioned earlier) is divided into 50 × 50 grids. That is, The spatial resolution is about 300 × 300 m. The aggregated data structure is shown in Fig. 3: the data is represented as several 1-hour time slices in the time dimension. Each time slice is defined as a 50 × 50 grid with 9 channels to reproduce the spatial relationship of the data. Each channel represents one of the above-mentioned 9 observed properties considered relevant to PM10.

After the aggregation, we consider two additional processing operations:

One processing operation is to slice and center the spatial grid. Precisely, we slice each of the above-mentioned 50×50 spatial grids, ensuring that the position we want to predict is in the center of the sliced grid. We're not sure if the model will benefit from this operation. According to Tobler's First Law of Geography [22], everything is related to everything else, but near things are more related than distant things. By slicing, we hopefully help the model exclude interference from distant noise. On the opposite side, in the SmartAQnet sensor network, some observed properties are only collected by a few sensors. An aggressive slicing may cause the input data to lose too much information about these observed properties, resulting in a dramatic decline in model performance. Therefore, we decided to demonstrate whether slicing should be performed through experimental results.

Another processing operation is to perform interpolation on the spatial grid. We think this helps eliminate heterogeneity in the data further. An obvious benefit is that even the most naive interpolation method can help the model distinguish whether an input 0 is a measured value of 0 or we don't know anything about it. Another potential benefit of interpolation is that it promises to alleviate the shortcomings of the slicing step mentioned above. By interpolation, we can generalize the information of observed properties recorded by only a few sensors to the entire grid, which leads to less information loss due to slicing. In addition, a "wonderful" interpolation that can take land use and the wind into account is expected to improve the homogeneity and expressiveness of the input data significantly. But since this problem belongs to another research direction, and its complexity is no less than the time series prediction problem, we plan to take this topic as a future research direction. In this article we only consider Inverse Distance Weighting (IDW) interpolation.

Window Generation. In this stage, we create time windows using the time slices generated in the previous temporal-spatial aggregation stage. After the window generation stage, a piece of training data for predicting the daily average PM10 concentration on day T looks as shown in Fig. 4. It consists of two parts: timestamp and time window.

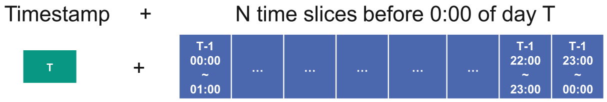


Fig. 4. A piece of training data processed after the window generation stage for predicting the daily average PM10 concentration on day T

For the timestamp part, we use two-dimensional relative timestamps. One dimension represents how many days have passed from the first day of this year until day T. Another dimension means which weekday day T is. We use relative timestamps because, during GPR training, we observed that GPR is difficult to give effective confidence interval estimates in extrapolation tasks. Because GPR does not find any experience from the training set in the corresponding area, thus it does not have any confidence in its prediction. We transform time series prediction into an interpolation problem using relative timestamps, resulting in more reliable predictions and confidence intervals. But this approach also comes at a cost. For example, it completely ignores the difference between different years. In fact, during the operation of the SmartAQnet project, we experienced the coronavirus pandemic. The lockdown policy is likely to lead to the data pattern over the years are not the same, thus affecting the model performance.

As for the time window part, we select N time slices before 0:00 of the day T, which were generated by the previous temporal-spatial aggregation stage. In our experimental setup, N takes a value of 24. That means the time window includes all the time slices of day T-1.

Feature Extraction. The feature extraction step is responsible for receiving the output of the data preprocessing step and performing feature extraction. Feature extraction refers to processing the features that need input to the model through methods such as screening or reorganization to eliminate information redundancy in the input data as much as possible.

In our model pipeline, the feature extraction step mainly has the following two contributions. First, feature extraction reduces the dimension of the input data, making the data more discriminable when the total amount of data is limited. Secondly, during the feature extraction process, the data will lose some unnecessary details (which usually could be treated as noise), which helps further to reduce the impact of data uncertainty on the model.

We tested three feature extraction methods during the model testing phase: Principal Component Analysis (PCA), Convolutional Neural Network (CNN), and Auto-Encoder. Among them, CNN and Auto-Encoder didn't perform well. We believe this is because our dimension reduction task is too heavy (the original input has about 540000 dimensions). At the same time, the total amount of data is too small (only c.a. 1500 available time windows established). PCA, on the other hand, performed well on data generated by all the above-mentioned preprocessing strategies. When set to capture 95% of the variance, PCA can reduce the preprocessed data from 540000 dimensions to 90—240 dimensions (depending on the settings of the preprocessing steps). In addition, PCA maintains some data interpretability. We finally decide to use PCA for feature extraction in the model pipeline.

Prediction Model. The prediction model is a machine learning model for regression tasks. As mentioned above, our pipeline uses a GPR-based prediction model.

Neural Kernel Network. The covariance function (kernel) is essential to the GP models. The choice of kernel determines almost all the generalization properties of a GP model. This is because the kernel incorporates prior assumptions about the characteristics of the data. In Vanilla GPR, the kernel selection relies heavily on the user's experience and prior knowledge of the data, but this is not always feasible. For example, users' prior knowledge of urban air quality prediction tasks is limited. Moreover, the data is often unrecognizable after many preprocessing steps. In addition, with the rise of combination kernel methods [7,11,12,21], it has been found that adding or multiplying multiple kernels can express more complex priors, which further increases the difficulty of choosing a proper kernel. In this context, the concept of compositional kernel learning is introduced. Its idea is to automate the selection and combination of kernels as part of the training process.

The Neural Kernel Network (NKN) [21] is a compositional kernel learning method. It uses a neural network-styled structure to represent the weighted addition and multiplication of kernels. And it can adjust the weights in the network through back-propagation to automatically select the structure of the combined kernel. The following Fig. 5 shows the basic structure of NKN.

First, we need to choose some commonly used kernels (such as RBF kernel, linear kernel, periodic kernel, RQ kernel, etc.) as the basic kernels, and these basic kernels are used as the input layer of the network. After that, each network layer can be divided into two parts, the first part is responsible for weighted addition, and the second is for multiplying adjacent results from the weighted addition step with each other. The last layer of the network has only one output, which can be seen as the result of the final combined kernel.

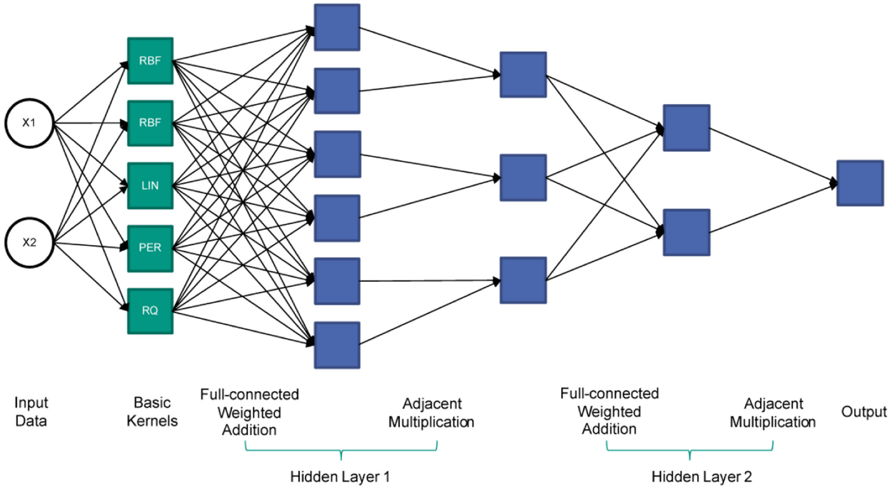


Fig. 5. The structure of a Neural Kernel Network kernel

Prediction Model Design. In our model pipeline, the prediction model consists of MLP and GPR (Fig. 6). We first use an MLP with 2 hidden layer to remap the output from the feature extraction step. Then the data will be input to a GPR model with a constant mean function and an NKN kernel for the regression task. We use RBF kernels, RQ kernels, Cosine kernels, and Matern kernels as the basic kernels of the NKN kernel. In the network part of the NKN kernel, we use two layers: the first layer has 4 outputs and the second layer has 1 output. In addition to this structure, we also implement some other regression models for performance comparison. Detailed information will be discussed in the following Sect. 4.

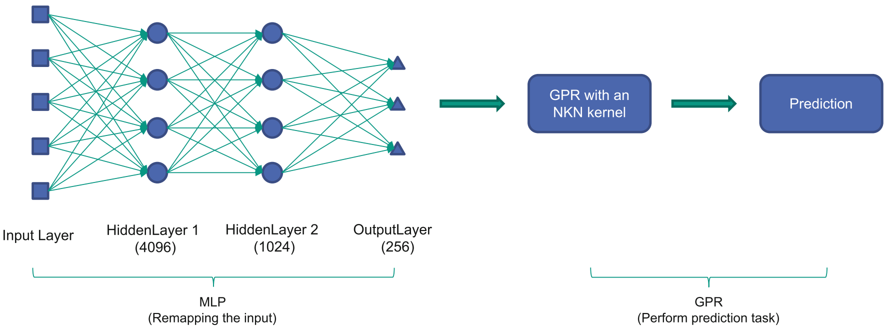


Fig. 6. The structure of the prediction model

4 Experiments

4.1 Model Performance and Comparison

We conducted controlled experiments to determine the optimal model pipeline steps and evaluate our predictive models' performance. As described in Sect. 3, three components need to be tested and assessed: whether interpolation is required, if grid slicing and centering are necessary, and which predictive model to use. For interpolation, we only consider two cases of no interpolation and IDW interpolation. For grid slicing and centering, we consider three cases: no slicing, big slicing (21×21 grid centered on the predicted point), and small slicing (11×11 grid centered on the predicted point). For the prediction model, in addition to the MLP + NKN kernel GPR proposed above, we also tested another five cases, namely MLP, LSTM, RBF kernel GPR, NKN kernel GPR, and MLP + RBF kernel GPR. That is, a total of 36 sets of experiments were carried out.

For each set of experiments, we first do a hyperparameter tuning. After that, we train 2 times for each of the 4 high-precision PM10 measurement stations and then average these 8 training results as the performance of this pipeline setting. We use this metric to decide which pipeline setting is the best for each prediction model. Then we train additional 3 times with the best setting of each model. We compared these results with each other and also with the baseline (using the previous day label as the predicted value). The result of the best performance of each prediction model and in which pipeline settings it was achieved is shown in the following Table 1.

From the results in Table 1, we can draw the following conclusions:

Table 1. The best performance of each prediction model

Model	Best MAE	Best PCC	Best Settings	
			Interpolation	Slice & Center
Baseline	4.71	0.502	-	-
LSTM	4.04 ± 0.07	0.575 ± 0.020	True	Big Slicing
MLP + RBF_GPR	3.89 ± 0.08	0.624 ± 0.015	True	Big Slicing
RBF_GPR	3.87 ± 0.06	0.635 ± 0.007	True	Big Slicing
NKN_GPR	3.80 ± 0.04	0.641 ± 0.006	True	Big Slicing
MLP + NKN_GPR	3.67 ± 0.08	0.665 ± 0.014	True	Big Slicing
MLP	3.66 ± 0.04	0.690 ± 0.001	True	Big Slicing

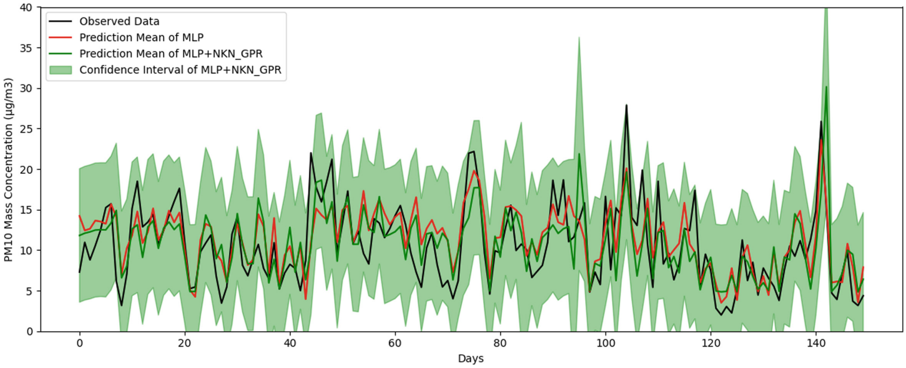


Fig. 7. MLP model (red line) can only give a single-valued prediction, GPR-based model (green line and interval), however, can also give a reasonable confidence interval of prediction through Bayesian theory. (Color figure online)

1. All prediction models give the best results in the experimental setting of interpolation + big slicing, which is consistent with our intuition described in Sect. 3.2.
2. Among all GPR-based models, the MLP + NKN_GPR model is the best-performing one. Overall, MLP achieves the best performance, and surprisingly, LSTM, which is usually considered a better solution to the time series problem, shows the worst result. After analysis, we believe this should be because the recurrent neural network (RNN) design determines that it is better at dealing with short-term dependencies. As an improvement to RNN, although LSTM has gained the ability to deal with long-term dependencies by adding gates mechanisms such as forget gates, the number of hidden units still limits its ability to express long-term memory. On the other hand, we believe that in this specific problem of PM10 forecasting, the short-term dependencies (such as the distribution over the last hour and the distribution of the same day in the previous week) and ultra-long-term periodic dependencies

(such as the same day of the other years) are dominant. In contrast, the more recent long-term dependencies (such as distributions from months ago), which LSTMs are good at handling, have relatively little impact on this problem. Furthermore, it must be pointed out again that MLP and LSTM also has the following shortcomings: MLP and LSTM are uninterpretable model and can only give a single-valued prediction, which means they cannot provide a reasonable confidence interval (Fig. 7, red line). In many scenarios (such as critical decision-making, when people are more reluctant to make mistakes), an uninterpretable single-valued prediction can only provide very limited help. The GPR-based model can give the confidence interval of prediction through Bayesian theory, which means the prediction given by the model is a Gaussian distribution. We can not only obtain the average value of this distribution (as a single-valued prediction) but also the standard deviation of this distribution can also be obtained (as the model's confidence in its predictions) (Fig. 7, green line and interval).

3. NKN kernel can effectively improve GP models' performance without prior knowledge of the dataset with the help of compositional kernel learning methods. Figure 8 shows the prediction results using the NKN kernel, the Matern kernel and the Cosine kernel. Since different kernels represent different prior assumptions about the dataset, their predictions are also entirely different. The NKN kernel can make multiple assumptions through its base kernels and then benefit from all these assumptions by learning the composition structure of these base kernels.

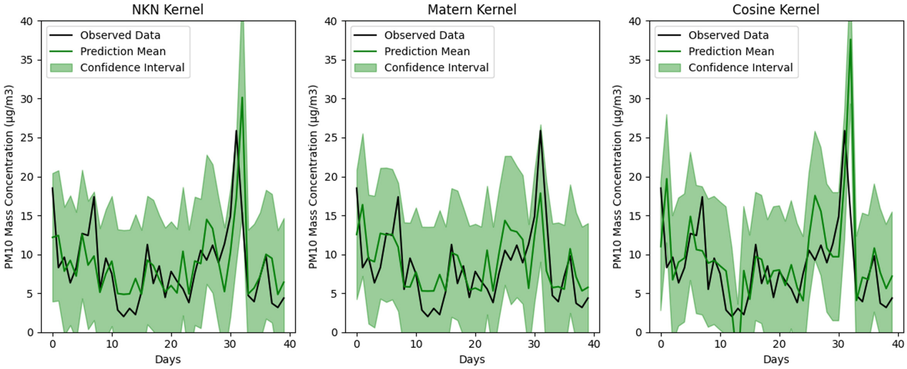


Fig. 8. Sample of the prediction results using the NKN kernel, the Matern kernel and the Cosine kernel

4.2 Evaluating the Role of Low-Cost Sensors in Prediction Tasks

As the first machine learning-based time series prediction study on the SmartAQnet dataset, we are also interested in the contribution that low-cost sensors can make to data analysis. Indeed, to improve sensor networks' temporal and spatial resolution, we must make a trade-off in cost. However, there have still been ongoing discussions about whether introducing ultra-low-cost sensors and citizen science projects into the network could benefit the datasets [3]. For this question, we also set up a set of controlled experiments.

The experimental setup is straightforward. We remove all data collected by ultra-low-cost sensors (84 CrowdSensing Nodes) from the dataset and then train the model pipeline with the same experimental setup as the above-mentioned best practice. It is worth noting that after removing these ultra-low-cost sensors, the entire sensor network still has over 100 sensors, which is still very dense for the spatial extent we model. We run each set of experiment 5 times, the following Table 2 shows the results of the experiments:

Table 2. The contribution that low-cost sensors can make to data analysis

	MAE ± std	PCC ± std
With Ultra-low-cost Sensors	3.67 ± 0.08	0.665 ± 0.014
Without Ultra-low-cost Sensors	4.14 ± 0.04	0.589 ± 0.032
Baseline	4.71	0.502

The results of the experiments are evident: although ultra-low-cost sensors introduce more heterogeneity and uncertainty into the aggregated dataset, they can significantly improve the expressiveness of the data when processed and analyzed appropriately.

5 Conclusions

In this paper, we propose a new model pipeline for time-series prediction of urban particulate matter based on heterogeneous sensor information. The model pipeline takes measurements aggregated from multiple internet sources with high heterogeneity and uncertainty as input and predicts the daily average PM10 mass concentration for the next day. We have experimentally determined the suitable components for the model pipeline: to sequentially perform temporal-spatial aggregation, spatial interpolation, slicing and centering, window generation, PCA dimensionality reduction on the input data, and then use MLP + GP regression with an NKN kernel to perform the prediction task. Ultimately, our model pipeline achieved an average MAE of 3.67 $\mu g/m^3$ and a Pearson correlation coefficient of 0.665.

Furthermore, we experimentally verify the contribution of citizen-run ultra-low-cost sensors in the prediction. Thanks to their meager cost, ultra-low-cost sensors can be large-scale deployed by institutional entities or popularized through citizen science programs. Although these sensors pose challenges such as heterogeneity and uncertainty, they significantly increase the temporal and spatial coverage of the data. We observe that the addition of ultra-low-cost sensor data reduces the average MAE of our prediction model from 4.18 $\mu g/m^3$ to 3.67 $\mu g/m^3$ and increases the PCC from 0.589 to 0.665. As long as processed and analyzed appropriately, ultra-low-cost sensors will definitely result in a significant performance improvement. We thus believe that post-hoc sensor networks that fuse different sensor sources in an opportunistic manner can thus advance knowledge in areas into which classical measurement systems cannot scale due to the cost of initial installation and maintenance. Prediction quality, however, can wildly vary based on the quantity and quality of the local sensor. Thus new prediction methods are needed that can derive certainty measures from the available data. We are confident that Gaussian process modeling can be a key component to inter- and extrapolating heterogeneous information sources. Machine learning approaches like neural networks can greatly help to derive fitting kernels that qualify relations between multiple heterogeneous data sources and choose hyperparameters according to the observed data.

Acknowledgments. We would like to give special thanks to Helmholtz European Partnership for Technological Advancement (HEPTA) for supporting this study. This work was also supported by Helmholtz AI Computing Resources (HAICORE) of the Helm-holtz Association's Initiative and Networking Fund through Helmholtz AI.

References

1. Bruns, J., Riesterer, J., Wang, B., Riedel, T., Beigl, M.: Automated quality assessment of (citizen) weather stations. GI-Forum **1**, 65–81 (2018)
2. Budde, M., et al.: Smartaqnet: remote and in-situ sensing of urban air quality. In: Remote Sensing of Clouds and the Atmosphere XXII, vol. 10424, pp. 19–26. SPIE (2017)
3. Budde, M., Schankin, A., Hoffmann, J., Danz, M., Riedel, T., Beigl, M.: Participatory sensing or participatory nonsense? mitigating the effect of human error on data quality in citizen science. Proc. ACM Interact. Mobile Wearable Ubiquitous Technol. **1**(3), 1–23 (2017)
4. Budde, M., et al.: Potential and limitations of the low-cost SDS011 particle sensor for monitoring urban air quality. ProScience **5**, 6–12 (2018)
5. Chen, S.: Beijing Multi-Site Air-Quality Data. UCI Machine Learning Repository (2019)
6. Cohen, A.J., et al.: Estimates and 25-year trends of the global burden of disease attributable to ambient air pollution: an analysis of data from the global burden of diseases study 2015. The Lancet **389**(10082), 1907–1918 (2017)
7. Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., Zoubin, G.: Structure discovery in nonparametric regression through compositional kernel search. In: International Conference on Machine Learning, pp. 1166–1174. PMLR (2013)
8. English, P., et al.: Performance of a low-cost sensor community air monitoring network in imperial county, CA. Sensors **20**(11), 3031 (2020)
9. Grothe, M., Broecke, J.V., Carton, L., Volten, H., Kieboom, R.: Smart emission-building a spatial data infrastructure for an environmental citizen sensor network. In: Jirka, S., Stasch, C., Hitchcock, A. (ed.) Proceedings of the Geospatial Sensor Webs Conference 2016 (GSW 2016), Münster, Germany, 29–31 August 2016, pp. 1–7. CEUR-WS.org/Vol-1762 (2016)

10. Hoffmann, B., et al.: Who air quality guidelines 2021-aiming for healthier air for all: a joint statement by medical, public health, scientific societies and patient representative organisations. Int. J. Public Health 88 (2021)
11. Jin, S.S.: Compositional kernel learning using tree-based genetic programming for gaussian process regression. Struct. Multidiscip. Optim. **62**(3), 1313–1351 (2020)
12. Kim, H., Teh, Y.W.: Scaling up the automatic statistician: scalable structure discovery using gaussian processes. In: International Conference on Artificial Intelligence and Statistics, pp. 575–584. PMLR (2018)
13. Kuremoto, T., Kimura, S., Kobayashi, K., Obayashi, M.: Time series forecasting using a deep belief network with restricted boltzmann machines. Neurocomputing **137**, 47–56 (2014)
14. Li, C., Budde, M., Tremper, P., Riedel, T., Beigl, M., et al.: Smartaqnet 2020: a new open urban air quality dataset from heterogeneous pm sensors. Proscience **8**, 1–10 (2021)
15. Li, J.J., Faltings, B., Saukh, O., Hasenfratz, D., Beutel, J.: Sensing the air we breathe-the opensense zurich dataset. In: Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
16. Martilli, A., et al.: Simulating the pollutant dispersion during persistent wintertime thermal inversions over urban areas. the case of Madrid. Atmos. Res. **270**, 106058 (2022)
17. Ong, B.T., Sugiura, K., Zettsu, K.: Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2. 5. Neural Comput. Appl. **27**(6), 1553–1566 (2016)
18. Qin, D., Yu, J., Zou, G., Yong, R., Zhao, Q., Zhang, B.: A novel combined prediction scheme based on CNN and LSTM for urban PM 2.5 concentration. IEEE Access **7**, 20050–20059 (2019)
19. Rivas, E., et al.: CFD modelling of air quality in Pamplona City (spain): assessment, stations spatial representativeness and health impacts valuation. Sci. Total Environ. **649**, 1362–1380 (2019)
20. Suganya, S., Meyyappan, T.: Adaptive deep learning model for air pollution analysis using meteorological big data. In: 2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4), pp. 1–6. IEEE (2021)
21. Sun, S., Zhang, G., Wang, C., Zeng, W., Li, J., Grosse, R.: Differentiable compositional kernel learning for gaussian processes. In: International Conference on Machine Learning, pp. 4828–4837. PMLR (2018)
22. Tobler, W.R.: A computer movie simulating urban growth in the detroit region. Econ. Geogr. **46**(sup1), 234–240 (1970)