An Explainable Deep Learning Architecture for Fine-Scale Air Quality Prediction Using Web Data

Anonymous Author(s)

ABSTRACT

Many scientific prediction problems have data- and modeling-related challenges, including handling of complex spatiotemporal variations, sparse ground truth measurements for validation, and identifying the main features driving the phenomena. This paper presents a novel and generalizable deep learning architecture that addresses these requirements and demonstrates the architecture with an important public health topic, air quality prediction, using easily accessible datasets on the Web. Reliable estimates of air pollutant concentrations, e.g., $PM_{2.5}$, over highly resolved spatial and temporal scales are critical to understanding the health effects. Existing statistical prediction models rely on expert-selected environmental factors (e.g., meteorological and geographic phenomena) and are typically tuned for specific study areas, pollutant types, and spatiotemporal scales. Machine learning methods reduce the need for domain experts but typically generate results that lack interpretability. In contrast, the proposed deep learning architecture can outperform state-of-the-art methods and generate interpretable results. The architecture includes a sparse layer, a convolutional, long short-term memory network (Conv-LSTM) with spatial filters of varying sizes, and a semi-supervised loss function. The sparse layer helps to identify important features affecting $PM_{2.5}$ from imbalanced environmental factors and provides model explainability. The Conv-LSTM simultaneously models the interactions between features over time and space at varying spatiotemporal scopes for $PM_{2.5}$ prediction. The semi-supervised loss function considers both labeled and unlabeled data in a neighborhood for overcoming the limitation of sparse sensors. During backpropagation, both supervised and unsupervised loss functions guide the updates of the entire network to 1) prevent overfitting, 2) refine feature selection, and 3) improve overall prediction. The experiment demonstrates that the proposed approach provides accurate fine-scale air quality predictions and reveals the critical environmental factors affecting the results.

KEYWORDS

$PM_{2.5}$, Spatiotemporal, Fine-Grained Prediction, Web Data

1 INTRODUCTION

Scientific prediction problems that deal with spatiotemporal phenomena (e.g., the prediction of traffic conditions, noise levels, and air quality) are challenging because they typically rely on data from only a few measurement locations (e.g., ground-based sensors) combined with auxiliary contextual data describing the surrounding environment (e.g., meteorological and geographic data). In order to generate interpretable results, spatiotemporal prediction algorithms need to take into account what are oftentimes complex spatial and temporal variations in addition to interactions in the measured data and with external contextual data. For example, in the public health domain, numerous epidemiological studies have shown strong associations between particulate matter air pollution with an aerodynamic diameter of fewer than 2.5 μm ($PM_{2.5}$) and a variety of health effects, including cardio-respiratory diseases and asthma (e.g., [5, 16]). However, relying solely on limited air quality sensors from fixed sites does not provide the spatial resolution required to characterize exposures where people spend their time: home, work and school. Having reliable ambient $PM_{2.5}$ exposure predictions (estimates) over a fine spatiotemporal resolution (e.g., 500 square meters) is of great importance for conducting health effects studies but is still an open research problem due to the difficulties in accessing contextual data (e.g., traffic), sparse real measurements of air quality, and the technical challenges in handling spatially and temporally correlated data (e.g., see [11, 12]).

In air quality prediction, typical sources of $PM_{2.5}$ measurements include: 1) remote sensing (satellite) measurements, which can provide daily, near-global coverage but are limited in that they do not directly represent ground-level concentrations of $PM_{2.5}$, requiring converting optical measures to a mass concentration. Furthermore, these data typically suffer from a high percentage of missing observations [9]; 2) ambient monitoring with gravimetric and beta-attenuation instruments, which provide high-quality measurements but are expensive to operate. As a result, the spatial distribution of such devices is usually sparse. For example, in the Los Angeles metropolitan area, the U.S. Environmental Protection Agency (US EPA) sets only nine monitoring stations (red stars in Figure 1); 3) mobile monitoring platforms (e.g., Google Street View cars) can be equipped with air quality sensors to collect street-level concentrations over large areas, but fleets are limited, and the data are not pervasive or accessible to the public; 4) low-cost sensors, such as PurpleAir,[1] provide a more widespread network that captures finer spatial and temporal variability than Federal- and State-operated monitoring networks [2]. Compared to the EPA monitoring stations, there are about 150 active PurpleAir sensors (purple dots in Figure 1) in the same area during November 2018. However, the spatial coverage still does not satisfy the needs of studies that require close tracking of exposure-response relationships (e.g., [15]).

To estimate air quality over highly resolved spatial and temporal scales given a set of fixed-site monitors, traditional spatial interpolation methods such as Inverse Distance Weighting (IDW) and ordinary Kriging, do not explicitly include explanatory variables about the environmental characteristics (from contextual data) such as meteorology and topography, which can limit their ability to produce reliable estimates [11]. In contrast, land-use regression (LUR) models generate explainable predictions of long-term spatial variations in air pollution levels by including land-use information (e.g., traffic indicators, industrial facilities, and population density) [6]. However, LUR models often rely heavily on empirical assumptions and expert-selected predictors and are not flexible enough to be generalized to different geographic regions [26]. More recently, machine

[1] https://www2.purpleair.com/

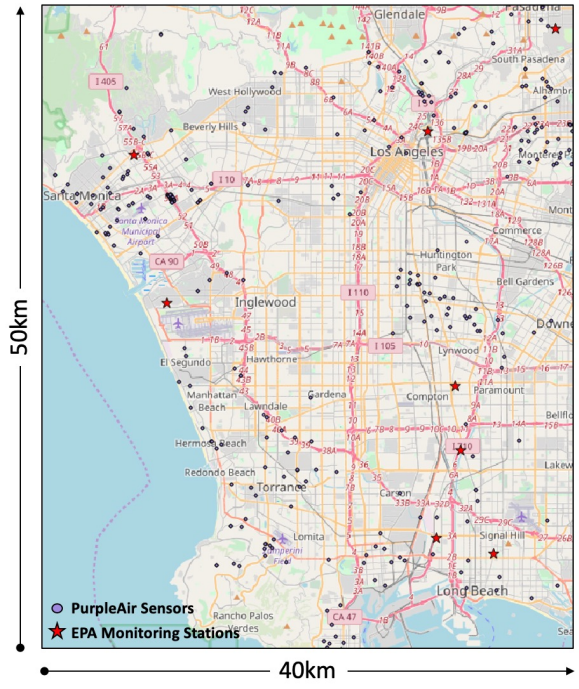


Figure 1: Red stars represent the locations of the EPA monitoring stations in the Los Angeles metropolitan area (more than 250km^2 per station). Blue dots represent the locations of PurpleAir sensors in the same region. The approximate geographical area size of the displayed map is 50km×40km.

learning-based approaches can deal with the air quality prediction problem from a data-driven perspective. In general, machine learning models aim to identify correlations between explanatory factors in the contextual data and the prediction (e.g., $PM_{2.5}$) automatically without domain knowledge [11].

Typically, fine-scale air quality prediction requires tackling several technical challenges. First is how to explicitly learn the most informative predictors from a variety of raw features that contextualize the environment (e.g., meteorologic and geographic features). Existing methods [4, 7] often directly adopt raw data on weather, road networks, and points of interest and hope the model can learn to select and use the best features. However, since these environmental features can have thousands of feature types and data ranges learning directly from the raw data, which might result in model overfitting, not generalizable over a large geographic region (e.g., spatial non-stationary issues), or poor predictive ability. Further, these models often fail to provide explainable results concerning environmental characteristics. The second challenge is how to jointly model multi-scale spatiotemporal effects on air quality. Zheng et al. [26] propose a method that trains separate classifiers for spatial and temporal features and combines the outputs with a co-training mechanism. However, separate models cannot effectively capture complex interactions such as emission patterns from neighboring factories that diffuse over space with high wind speed in short time frames. The third challenge is how to overcome the limitation of sparsely labeled locations. Only leveraging limited labeled locations might ignore the spatial dependency among the large number of unlabeled locations, which might lead to inaccurate predictions.

This paper proposes a deep learning architecture for fine-scale predictions of location-dependent time series data and applies the approach for air quality prediction using publicly available and easily accessible Web data, including geographic data for describing the nature and built environment from OpenStreetMap (OSM),[2] air quality data from PurpleAir, and meteorological data from DarkSky.[3] These data either have global coverage (OSM and DarkSky) or alternative data exist in many places (e.g.,AirBox for air quality data from low-cost sensors [3]). Predicting air quality using Web data that have global coverage or are easily substituted with other similar data has the advantage that the same approach can apply to many places without retooling the architecture to accommodate the availability of a dataset. For example, air quality data from ambient monitoring sites and low-cost sensors are usually available online and easy to access while remote sensing data and street view data are proprietary, which are sometimes costly and challenging to obtain.

Our approach first generates a high-resolution grid map (covering the target area), in which each cell contains air quality labels (if the cell covers a sensor), dynamic features (e.g., weather conditions), and static features (e.g., topographic information). Because the number of raw features can be large and imbalanced, irrelevant features can increase noises and computational cost. Our model first embeds a single-connected feature selection layer that encourages the parameters to be sparse using the L_1 regularization. The sparse layer not only automatically selects relevant features (e.g., wind speed and primary roads) but also reduces training data noise by removing the features with small weights. The selected features and their weights can be used to explain air quality-related factors. After feature selection, our model uses an auto-encoder to learn the latent embedding of both dynamic features and static features. Here, the auto-encoder is responsible for learning the joint effects between the features as well as reducing the dimension of the feature vector. Next, our model inputs the feature representations to multiple Conv-LSTM layers [23] with varying kernel sizes to learn their contributions at various spatial distances. The outputs of the Conv-LSTM layers are concatenated to feed in the fully connected layers to generate predictions on the grid map. In contrast to the existing methods that separately handle spatial and temporal components, the Conv-LSTM layer leverages convolution operation to capture the spatial relationship directly in the recurrent neural network and simultaneously model the effects from space and time. Finally, since the training data only contain limited labeled locations (i.e., sparse air quality sensors), the network might not be able to learn the diverse environmental factors and result in overfitting. Our approach takes advantage of the plentiful unlabeled locations to overcome this challenge by introducing a spatially constrained semi-supervised loss function to reduce the variation between nearby locations. By connecting the above layers, our model can jointly learn spatiotemporal effects to produce accurate and smooth air quality predictions at a high resolution with automatically-selected features.

[2] https://www.openstreetmap.org/
[3] https://darksky.net/dev

The main contribution of this paper is a general network architecture for fine-scale predictions of location-dependent time-series data from limited labeled data and its application of an air quality prediction model using Web data. The specific contributions of the proposed network architecture and its application are to: 1) automatically extract informative predictors among a variety of raw features from multi-source data, 2) jointly model spatiotemporal effects to generate accurate fine-scale predictions, 3) effectively leverage the information from unlabeled locations to overcome the limitation of sparse training data, and 4) predict air quality using publicly available data from the Web. We demonstrate our proposed model in the air quality domain, but the architecture is generalizable and can be applied to handle other complex scientific prediction problems involving location-dependent time-series data (e.g., traffic).

2 SPATIOTEMPORAL PREDICTION MODEL

This section presents the architecture for fine-scale air quality prediction using Web data. The goal is to predict air quality values ($PM_{2.5}$ concentrations) on a fine-scale (high-resolution) spatial grid (e.g., cell size of 500 meters by 500 meters, 500m×500m). A 3D tensor $\boldsymbol{X} = (F, H, W)$ represents the input grid data, where F is a set of features (ground truth measurements and contextual data), and H and W are the height and width of the grid, respectively. The output tensor is $\boldsymbol{Y} = (P, H, W)$, where P represents the predictions (dimension=1). Let $\boldsymbol{X}^{(t)}$ represent the input signal at time t, T' is the number of previous hours (i.e., from $t - T' + 1$ to t). The proposed architecture aims to learn a function h that maps T' historical input signals to the output at t:

$$[\boldsymbol{X}^{(t-T'+1)}, \cdots, \boldsymbol{X}^{(t)}] \xrightarrow{h} [\boldsymbol{Y}^{(t)}]$$

Figure 2 shows the proposed architecture. After constructing the input spatial grid (Section 2.1), the model embeds a sparse layer for automatically selecting air quality-related predictors (Section 2.2) from a large set of features. Next, the model utilizes an encoder-decoder module to learn distinguishable interactions of the selected features (Section 2.3). Afterward, the model uses the Conv-LSTM layers with varying kernel sizes to jointly learn the spatiotemporal impacts on air quality from multiple spatial and temporal distances (Section 2.4). The model then concatenates and feeds the outputs to fully connected layers to generate the prediction grid. Finally, the model employs a semi-supervised loss function to enforce the spatial dependency of the generated predictions by constraining the variation between nearby locations and close time points. The architecture does not assume a specific application.

2.1 Generating Grid Data

The raw data (e.g., PurpleAir sensor data and contextual data) can have varying spatial and temporal resolutions. To unify and aggregate the input data, the system divides the target area into disjointed cells and transforms the raw data into a grid representation. Each cell in the grid represents a specific area on Earth. Each cell contains a set of features, F, which represent an aggregation of the contextual data. We denote the feature vector for a cell as $F = [F_d, F_s]$, consisting of dynamic (time-varying) features, F_d (e.g., weather) and static (time-invariant) features, F_s (e.g., built environment). If the observations of a feature type are sparser than the grid (target) resolution but uniformly distributed in space (e.g., weather data from DarkSky), the system spatially up-scales the feature using the cubic interpolation method; otherwise, the system either directly adopts the contextual data value to generate the feature vector of a cell or aggregates the contextual data values of a feature type (e.g., the number of hospitals or commercial buildings) within the cell to generate the feature vector [11]. Figure 3(a) shows an example of one feature component, primary roads, in the spatial grid covering Los Angeles. This primary road component is a summation of the road lengths of individual primary roads in each cell.

This process assumes that the air quality value in a cell is uniform (i.e., one prediction per cell). The system maps the available sensors to the corresponding grid cells to generate prediction labels. If one cell contains multiple sensors, the cell contains the average of their sensor readings. Figure 3(b) shows an example of a $PM_{2.5}$ grid map. The colored cells are locations with ground truth measurements (i.e., labeled locations) while the uncolored cells are the target (unlabeled) locations for prediction. The goal is to predict the $PM_{2.5}$ concentrations for every cell in the $PM_{2.5}$ grid map.

2.2 Feature Selection

The proposed network architecture embeds a sparse layer (Figure 2, "Sparse Layer") for feature selection with the purpose of (1) reducing noise by removing irrelevant input features, and (2) explaining the results with the selected features. The sparse layer is a linear layer containing the same number of nodes as the input, and there exists only a single connection between the corresponding nodes. We denote the weight matrix of the sparse layer as $W^{(sp)}$, which is a diagonal matrix (zero off-diagonal weights).

To perform feature selection, we add L_1 regularization as the sparse constraint on this layer. L_1 regularization utilizes a penalty term that forces the sum of the absolute values of the parameters to be small. Thus, L_1 regularization can cause many weights to be close to zero. If the weight is a small value (less than a predefined threshold), our model removes the corresponding feature from the network (set to zero) to achieve the purpose of feature selection. The cost function of L_1 regularization is as follows:

$$L_{sp} = \sum_{w \in W^{(sp)}} ||w|| \tag{1}$$

2.3 Learning Feature Representations

The proposed network architecture leverages an auto-encoder to learn a new feature representation from F_d and F_s after feature selection. F_d and F_s have three significant differences: (1) F_d are changing along the time dimension while F_s are constant, which means F_s repeats for each time point; (2) the length of F_d (i.e., the number of feature types) is much less than F_s, so the impacts of F_d in the network might be overwhelmed by F_s; (3) F_s are usually sparse (having many zeros). For example, there might be 300 different types of geographic features for describing the built environment, but each cell might contain only a few of them (e.g., a cell only contains industrial lands and nothing else). Therefore, it is necessary to learn

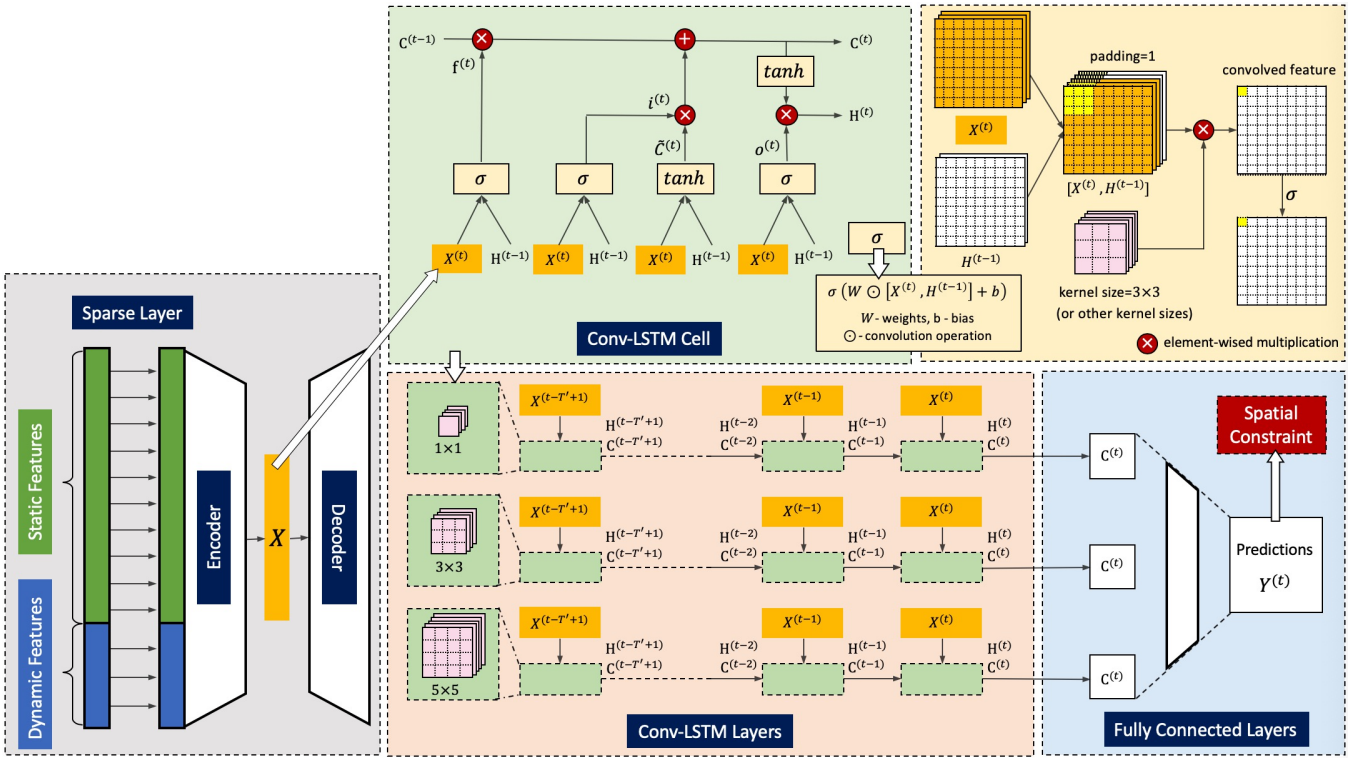


Figure 2: Overall network architecture

a latent embedding to condense these two types of features instead of directly feeding the original feature vector to the network.

Auto-encoder is an unsupervised learning technique for the task of representation learning. The auto-encoder tries to learn a function, $h_{w,b}(x) = \hat{x} \approx x$, by minimizing the reconstruction error between $\hat{x}$ and x. The middle layer serves as the latent embedding of the inputs. By embedding an auto-encoder layer (Figure 2 "Encoder" and "Decoder"), the network can generate a new representation capturing the joint effects of F_d and F_s. For example, suppose a primary road is an indicator of traffic volume, and its impact can vary according to the time and weather conditions. Another benefit is that the encoded feature vector is smaller than the original vector, which reduces the number of parameters in the network. The proposed system pre-trains the auto-encoder on the raw input vector (before feature selection) and uses the encoder as the second component in the network after the feature selection layer. During the training process, the network continuously updates the parameters of encoder and decoder for fine-tuning. The reconstruction loss is as follows:

$$L_{ae} = \bar{L}(x_{sp}, \hat{x_{sp}}) \tag{2}$$

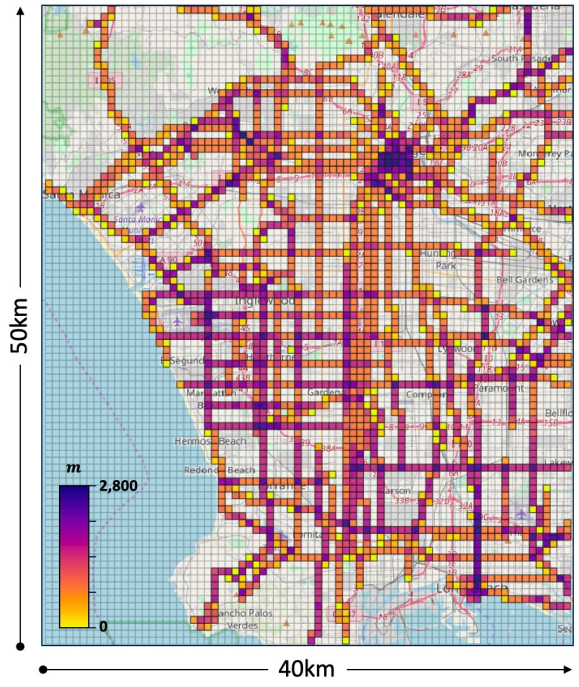
where x_{sp} is the output of the sparse layer (the encoder input), $\hat{x_{sp}}$ is the reconstructed vector (the decoder output), and $\bar{L}$ is the loss function using the mean square error.

2.4 Learning Spatiotemporal Impacts

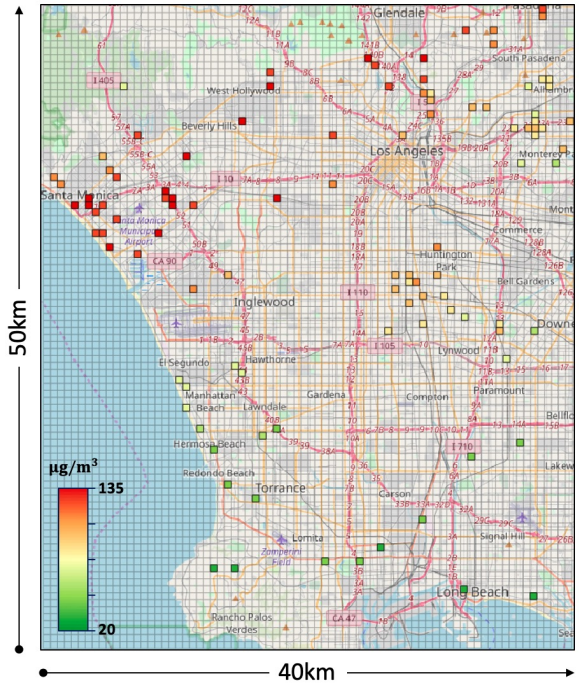
The current air quality level is highly correlated with the environmental characteristics at present and also from previous time points and neighboring locations. We define spatiotemporal impacts as the joint effects of air quality-related factors from space and time. For example, suppose a South-West power plant emits pollution at time T (Figure 4(a)), the North-East cells can be significantly polluted at $T + 1$ with a North-East wind direction (Figure 4(b)).

For modeling the spatial dependency, convolution operation can extract important salient information from neighboring pixels in image classification and recognition tasks [22]. One advantage of using convolution operations for location-dependent data is that it retains the positional relationships between data cells. For example, the influence on the left cell on the center cell should be heavier than the influence on the right cell when the wind direction is to the left. For modeling the temporal dependency, recurrent neural network (RNN) performs better than traditional models by automatically computing the information passing to the next time step in the sequence and using the final stored memory to make decisions (e.g., [12]).

To jointly model the spatiotemporal impacts on air quality, the proposed architecture leverages the Conv-LSTM operation [23] that adds the convolution operation directly in the recurrent neural network. Here, the constructed grid feature map can be directly treated as an image with multiple channels. For example, in Figure 5, when predicting the air quality value for the center cell (red box) by considering one-step neighbors (in the purple dotted box), the model should learn the interactive effects from the North-East green areas, the North-West residential areas, the South-East industrial areas, and the East commercial areas. In practice, the model learns useful information from the combination of the current latent embedding

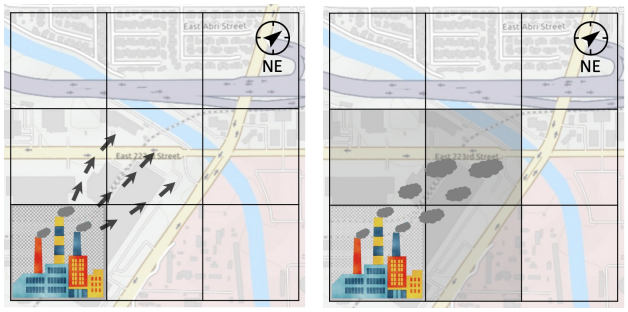


(a) Grid primary roads



(b) Grid $PM_{2.5}$ Concentrations

Figure 3: An example map with 500m×500m cells covering a 50km×40km area in Los Angeles (a) Aggregated length of primary roads in each cell: The darker the color, the more primary roads in the cell. Missing values are not colored (b) $PM_{2.5}$ concentrations after mapping sensor data to the grid.



(a) Time T (b) Time $T+1$

Figure 4: An example of spatiotemporal effects. The cell size is 500m×500m. (a) A pollution emission from South-West power plant at time T (b) The North-East cells can be polluted at time $T+1$ with the North-East wind.

(i.e., the interactive effects) of the selected features from neighbors and the previous hidden memory in the Conv-LSTM Cell (Figure 2). The Conv-LSTM operation replaces the internal matrix multiplications of the original LSTM with convolution operations using the following equations:

$$\begin{aligned}
\boldsymbol{i}^{(t)} &= \sigma(\boldsymbol{W_i} \odot [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b_i}) \\
\boldsymbol{f}^{(t)} &= \sigma(\boldsymbol{W_f} \odot [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b_f}) \\
\widetilde{\boldsymbol{C}}^{(t)} &= \tanh(\boldsymbol{W_C} \odot [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b_C}) \\
\boldsymbol{C}^{(t)} &= \boldsymbol{f}^{(t)} \times \boldsymbol{C}^{(t-1)} + \boldsymbol{i}^{(t)} \times \widetilde{\boldsymbol{C}}^{(t)} \\
\boldsymbol{o}^{(t)} &= \sigma(\boldsymbol{W_o} \odot [\boldsymbol{X}^{(t)}, \boldsymbol{H}^{(t-1)}] + \boldsymbol{b_o}) \\
\boldsymbol{h}^{(t)} &= \boldsymbol{o}^{(t)} \times \tanh \boldsymbol{C}^{(t)}
\end{aligned}$$

where $\boldsymbol{i}$, $\boldsymbol{f}$, $\boldsymbol{C}$, and $\boldsymbol{o}$ correspond to the input gate, forget gate, memory cell, and output gate in the LSTM, respectively; $\boldsymbol{X}^{(t)}$ is the current input, and $\boldsymbol{H}^{(t-1)}$ is the last output; $\odot$ denotes the convolution operation, $\times$ is the element-wised multiplication; $\boldsymbol{W}$ terms denote the weight matrices, and $\boldsymbol{b}$ terms are the bias vectors.

To investigate the influence of environmental characteristics on air quality from varying distances, the proposed architecture employs multiple Conv-LSTM layers with various kernel sizes to learn the impacts from the neighbors within different distance ranges. For example, if the kernel size is three and the cell size is 500m×500m, the model looks at one-step neighbors within approximately 1,500m. Figure 2 shows an example of leveraging three kernel sizes in the "Conv-LSTM Layers". The outputs of multiple Conv-LSTM layers are concatenated and fed to the fully connected layers to generate air quality predictions for each cell.

2.5 Semi-Supervised Loss

Since the number of unlabeled cells is much greater than the labeled cells, the proposed architecture incorporates two types of losses in a semi-supervised way. First, the network calculates the supervised loss using the following cost function:

$$L_{pred} = \sum_{i=1}^{m} \bar{L}(p_i, q_i) \tag{3}$$

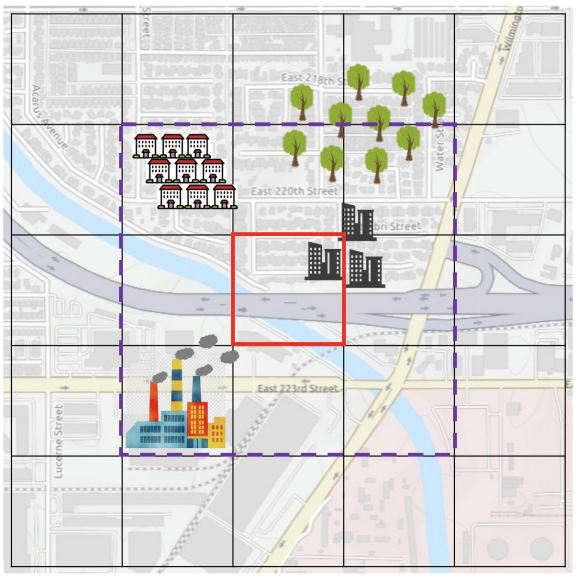


Figure 5: An example of learning spatial effects from one-step neighbors effects. The cell size is 500m×500m. The red box is the target cell and the purple dotted box contains environmental characteristics of one-step neighbors.

where m is the number of labeled cells, p is the prediction value, q is the real observation, and $\bar{L}$ is the loss function with the mean square error (MSE).

L_{pred} depends only on the labeled cells, which might cause overfitting because of the limited number of sensors. Due to the relatively small cell size in the fine-scale prediction task, the proposed architecture employs a spatial constraint assuming air quality values at close spatial distance tend to be similar. The proposed architecture applies the following semi-supervised cost function on the predictions (of both labeled and unlabeled locations) to force nearby cell values to be similar:

$$L_{sc} = \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j \in N_{k_i}} e^{-k} \bar{L}(p_i, p_j) \tag{4}$$

where n is the number of total cells, K is the number of neighboring steps, N_{k_i} is the neighboring cells of cell i within step k, and $\bar{L}$ is the loss function with MSE.

The overall cost function of the proposed architecture is the sum of equation (1), (2), (3), and (4):

$$L_{loss} = L_{pred} + \alpha \times L_{sp} + \beta \times L_{ae} + \gamma \times L_{sc} \tag{5}$$

where L_{pred} is the loss over the training examples, L_{sp} is the loss from the sparse layer, L_{ae} is the reconstruction loss of the auto-encoder, L_{sc} is the loss from applying the spatial constraint; α, β, and γ are hyper-parameters. We train our model by updating the network parameters using (5) with backpropagation.

3 EXPERIMENTS

We implemented the proposed network architecture with Python 3.6 and Pytorch framework. We conducted the experiments in a Docker container deployed on a GPU server with four physical cores and 64GB memory. The spatial computing processes were done in PostGIS (e.g., gridding and spatial aggregation).

3.1 Experimental Settings

3.1.1 Datasets. We leveraged the following datasets covering a 50km×40km region in Los Angeles county (Figure 6(a)):

(1) Air quality data: Our system collected hourly $PM_{2.5}$ concentrations for November 2018 from 165 PurpleAir sensors. PurpleAir sensors provide measurements of several particulate air pollutants including $PM_{1.0}$, $PM_{2.5}$, and PM_{10} (in units of $\mu g/m^3$) and are classified as "outdoor", "indoor", and "None". Our system only utilizes "outdoor" $PM_{2.5}$ measurements.
(2) Meteorological data: Our system collected hourly weather data for November 2018 from DarkSky. DarkSky reports worldwide fine-scale weather data on a variety of features, of which we extracted eight (temperature, dew point, humidity, pressure, wind speed, wind direction, visibility, and cloud cover). The resolution of the requested weather data is of 5km×5km, so our system interpolates them to the target resolution.
(3) Geographic data: Our system extracted geographic information from OpenStreetMap, to describe land use, roads, traffic, railways, and water areas in various spatial representations (polygons, lines, and points). Each geographic feature contains various sub-types; for example, "roads" are classified into motorways, primary roads, residential roads, etc.
(4) Other data: Our model generated the time information (i.e., hour of a day, day of a week, and day of a year) as additional dynamic features. Besides, our model added the coordinate of the grid center (i.e., longitude and latitude) in the feature vector to represent the location information.

3.1.2 Training Settings. We split the sensor locations into training (60% of the available locations), validation (20%), and testing locations (20%). We randomly repeated this splitting process three times to obtain an average evaluation score. To evaluate our prediction model on multiple spatial resolutions, we created two separate gridded surfaces over the same region with cell sizes of 500m×500m and 1,000m×1,000m. Table 1 shows the details of the air quality datasets after mapping air quality sensor data to the grid maps.

To predict $PM_{2.5}$ concentrations at time t, we generated the input features with temporal lags of the previous 12-hours (i.e., from $t-11$ to t). We set the number of layers in the auto-encoder to 6 with the latent embedding of 16 neurons. We constructed multiple Conv-LSTM layers with 64 cells in one hidden layer. We applied three kernel sizes (1×1, 3×3, and 5×5) for the 500m×500m cells, i.e., at most two-step neighbors (distance range = 1,500m). For the 1,000m×1,000m cells, we applied two kernel sizes (1×1 and 3×3) for capturing neighboring information within 2,000m. We set the learning rate for the training process as 0.001. We set neighboring step, K, as 1. α, β, γ are the feature selection parameter, the reconstruction loss parameter, and the spatial constraint parameter, respectively.

Table 1: Details of $PM_{2.5}$ grid data

Resolution	500m×500m	1,000m×1,000m
Number of Labels	138	126
Number of Grids	6,992	1,748
$PM_{2.5}$ Average	25.99	26.01
$PM_{2.5}$ STD.	27.40	27.38

3.1.3 Baseline Methods. We compared the results from our proposed model to the following approaches:

(1) Random Forest (RF): Yu et al. [25] demonstrated that RF outperformed other machine learning approaches including logistic regression, decision tree, and artificial neural network for air quality prediction. Therefore, RF is selected as one of the baseline methods.
(2) Inverse Distance Weighting (IDW): IDW is a standard spatial interpolation method that calculates the weighted average of the air quality observations from all sensors for the target location L at time T with the following equation:

$$Pred_{L,T} = \frac{\sum_{i=1}^{n} s_i \times w_i}{\sum_{i=1}^{n} w_i}$$

where n is the number of sensors, s_i is the $PM_{2.5}$ value for the i^{th} sensor at time T, and w_i is the inverse spatial distance between the target location and the i^{th} sensor.
(3) LSTM: The LSTM model generates predictions by learning previous 12-hour information for individual locations. The model does not consider spatial effects.
(4) Conv-LSTM: The Conv-LSTM model is used to demonstrate the benefit of feature selection and the spatially constrained semi-supervised loss function in the proposed architecture.

3.1.4 Evaluation Metrics. For the cell locations where labeled data (real observations of air quality) are available, we used the following metrics to evaluate model performance:

(1) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}}$$

(2) R-Squared (R^2) provides a measurement of how well the model fits the testing data or how much variability in $PM_{2.5}$ can be explained by the model. We computed the R^2 score using *R2-score* in Python scikit-learn.

For the unlabeled locations and model explainability, we verify the spatial variation of the predicted value with the environmental characteristics with air quality studies in the literature.

3.2 Results and Discussion

3.2.1 Model Performance. Our deep learning model outperformed all other approaches at both 500m×500m and 1000m×1000m resolutions for predicting $PM_{2.5}$ concentrations (Table 2). We observe that RF performed poorly with high RMSE and low R^2 because it cannot handle complex relationships between the predictors adequately. Conv-LSTM achieved much lower RMSE and higher R^2 than LSTM, demonstrating that incorporating spatial effects significantly improved model performance. We observe further performance gains with our model, which outperformed Conv-LSTM by adding sparse and spatial constraints. This demonstrates that removing irrelevant features and modeling spatial dependency with unlabeled data helps to reduce model uncertainty.

Our model generated hourly $PM_{2.5}$ concentration predictions ($\mu g/m^3$) for the target region (Figure 6(a)) in November. Figure 6(b) and Figure 6(c) present the aggregated prediction results (monthly average) at the 1,000m×1,000m resolution using our model and IDW. Figure 6(d) and Figure 6(e) show the results at the 500m×500m resolution. We can see that the average predictions from IDW were smooth over the region, which only offer a general idea about the variation of the $PM_{2.5}$ concentrations at a coarse spatial scale (e.g., the Los Angeles downtown area has poorer air quality than other areas). Moreover, IDW produced some odd "bull's eye" around some data locations due to the uneven distribution of sensors. In comparison, our predictions provided additional spatial detail, consistent with some of the selected features. For example, $PM_{2.5}$ concentrations along freeways (I-110, I-10, I-5, and I-405) were clearly observed and higher due to traffic-related sources prevalent in Los Angeles [20]. Also, the Southern coastal region (Long Beach) experienced higher $PM_{2.5}$ concentrations due to heavy industrial emissions around the ports of Los Angeles. By comparing the results between two resolutions, we can see that the grid maps show clear impacts of environmental characteristics on air quality when the resolution is higher, which might be because 1,000m×1,000m cell size is too large for showing the spatial effects. In general, our model successfully generated predictions based on the selected relevant predictors, which explains how the $PM_{2.5}$ concentrations vary with environmental characteristics over the space.

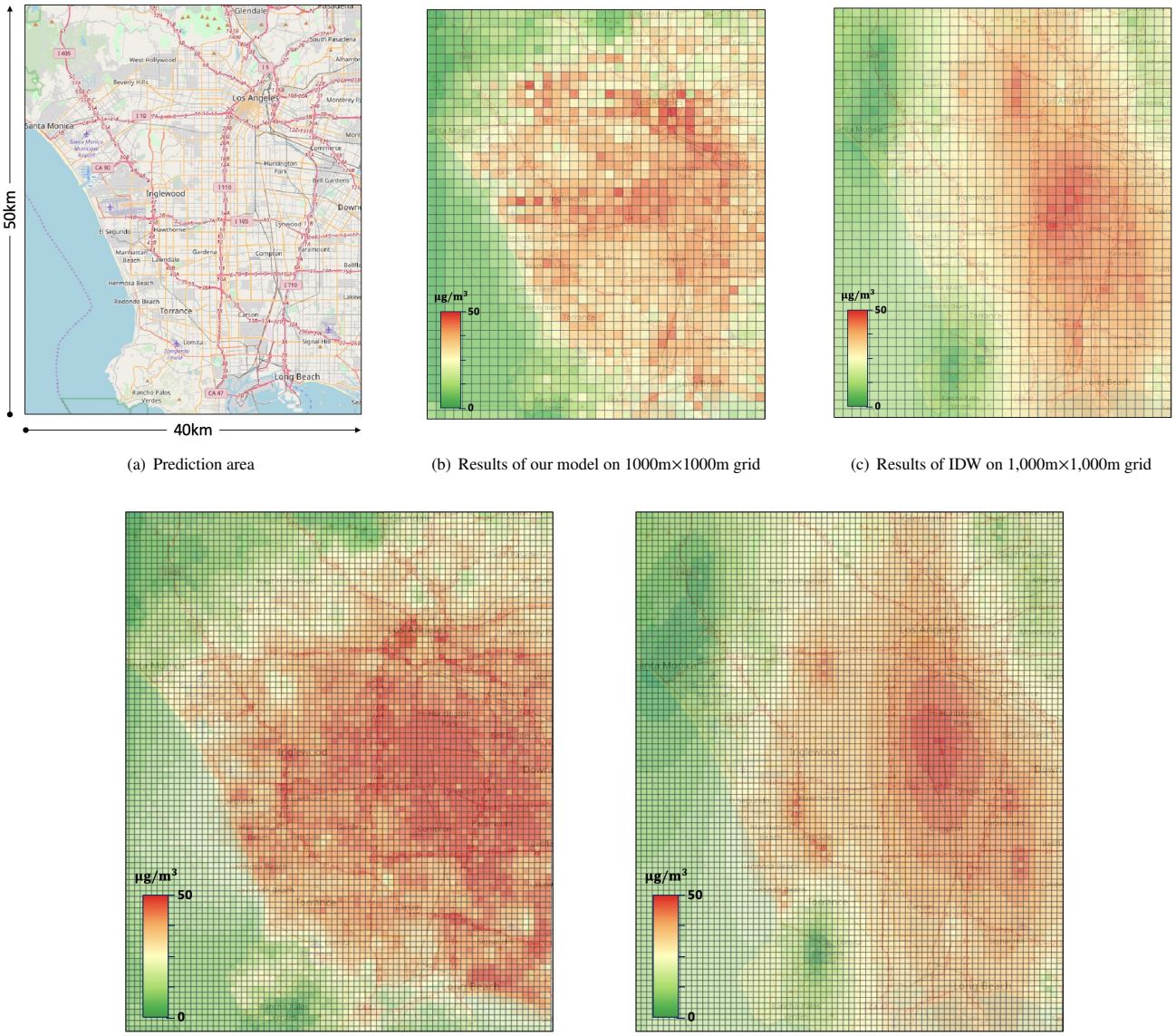
Figure 7 shows the time series comparison between our $PM_{2.5}$ predictions and the observed sensor concentrations at two sample testing locations. Our model performed well at most time points but inclined to underestimate at times where a sharp increase occurred. For example, the observed peak between November 10^{th} and November 11^{th} was due to the Woolsey fire.[4] Both examples in Figure 7 show this peak, but our predicted concentrations are 10-20 $\mu g/m^3$ less than the sensor measurements.

Table 2: Comparison between our model and the baselines

	500m×500m		1,000m×1,000m	
Method	**RMSE**	**R2**	**RMSE**	**R2**
RF	14.184	0.739	14.424	0.729
IDW	9.851	0.872	10.754	0.847
LSTM	15.620	0.630	15.032	0.641
Conv-LSTM	11.586	0.775	11.303	0.837
Our Model	**9.704**	**0.876**	**10.256**	**0.861**

3.2.2 The Impact of Feature Selection. We describe the number of features selected by our model and the corresponding RMSE as a function of α in Figure 8 by fixing $\beta = 10$ and $\gamma = 1$. We observe

[4] https://ktla.com/2018/11/11/unhealthy-air-plagues-much-of-southern-california-for-another-day-as-woolsey-fire-burns/



(a) Prediction area

(b) Results of our model on 1000m×1000m grid

(c) Results of IDW on 1,000m×1,000m grid

(d) Results of our model on 500m×500m grid

(e) Results of IDW on 500m×500m grid

Figure 6: (a) shows the target prediction region covering an area of 50km×40km. (b)(c)(d)(e) are the aggregated $PM_{2.5}$ concentration predictions in November.

that the number of selected features decreases as α increases, which demonstrates the effectiveness of the feature selection layer. When α is less than 2, the model fails to select feature correctly, which results in a high RMSE. In comparison, when α is continuously increasing, the RMSE increases as the number of features decreases, which could be because some useful information is also filtered by the sparse constraint. Therefore, we set α as 2.

The input feature vector consists of a total of 11 dynamic features (8 meteorological features and 3 time features) and 82 static features (80 geographic features and 2 topological features). We set the threshold as (-0.001, 0.001) and removed features from the network if their weight in the sparse layer was within this range. Table 3 shows the selected dynamic features and Table 4 are the selected static features for predicting the grid map of resolution 500m×500m (sorted by weights). The listed weight is the absolute value of the real weight in the sparse layer for the corresponding feature. We can see that all dynamic features were automatically chosen by our model with relatively higher weights than the selected static features. The strong relationship between $PM_{2.5}$ concentrations and "visibility" is expected and has been seen previously [17], which

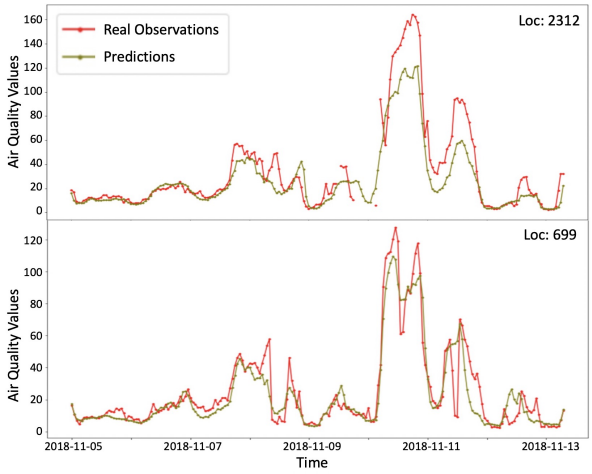


Figure 7: $PM_{2.5}$ Predictions for November 5^{th} to 13^{th}, 2018

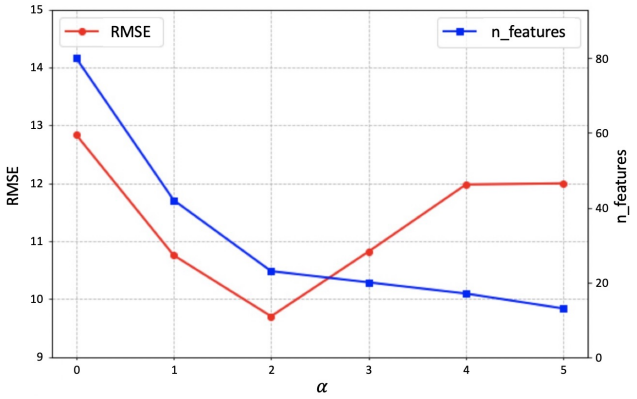


Figure 8: The number of selected features and the corresponding RMSE as a function of α

Table 3: Selected dynamic features

Dynamic Feature (weight)	Dynamic Feature (weight)
visibility (0.3833)	cloud_cover (0.1713)
pressure (0.2919)	temperature (0.1393)
wind_speed (0.2118)	humidity (0.0880)
hour_of_day (0.2106)	day_of_week (0.0880)
dew_point (0.1896)	wind_direction (0.0725)
day_of_year (0.1713)	

Table 4: Selected static features

Static Feature (weight)	Static Feature (weight)
latitude (0.1859)	landuse_a_industrial (0.0501)
longitude (0.0879)	roads_primary (0.0484)
railways_rail (0.0754)	roads_secondary (0.0408)
traffic_fuel (0.0712)	landuse_a_farm (0.0242)
roads_motorway_link (0.0676)	waterways_river (0.0200)
railways_light_rail (0.0646)	traffic_turning_circle (0.0176)
traffic_stop (0.0505)	roads_pedestrian (0.0161)

is consistent with our autonomous selection. Megaritis et al. [13] demonstrated $PM_{2.5}$ concentrations are highly related to "temperature", "humidity", "pressure", and "wind speed", which were also automatically selected by our model. Time information including weekday/weekend and daytime/night is important since it indicates temporal variation and population mobility. In addition, the model selected 12 static geographic features plus longitude and latitude. Most of the selected features are consistent with existing studies on $PM_{2.5}$ in Los Angeles. For example, Kam et al. [8] demonstrated that the light-rail lines and subways are strongly associated with ambient PM levels in Los Angeles (R^2=0.61) by personally monitoring the air quality at the stations. Moore et al. [14] showed that industrial areas, arterial roads, open areas are statistically significantly associated with $PM_{2.5}$ in Los Angeles (R-value is approximately 0.4 to 0.6 respectively) using LUR approach. These existing studies required long-term and costly investigations by the environmental scientists with domain-specific expertise. In contrast, our method automatically learns from a variety of raw features, selecting those that are most pertinent to air quality prediction. More importantly, the selected predictors are consistent with existing literature on PM exposure assessment in Los Angeles. For other selected static features, such as "traffic_fuel" (referring to gas stations) and "traffic_stop", we did not find the existing work on analyzing the relationship between them and $PM_{2.5}$ concentrations to demonstrate our results. However, the "traffic_fuel" and "traffic_stop" features intuitively relate to vehicle air pollution emissions. Furthermore, these features can vary significantly from one study location to another so autonomous selection is important for reducing expert intervention for feature selection. Thus, our model is applicable to other regions that cannot afford these longitudinal studies and offer explainable prediction results. By using only Web data that are available to the public, our approach eliminates the needs for purchasing expensive datasets (e.g., historical and real-time traffic data) or installing additional sensor hardware for a study location.

3.2.3 The Impact of Auto-Encoder. Figure 9 shows the accuracy (RMSE) as a function of β by fixing $\alpha = 2$ and $\gamma = 1$. We can see that our model achieved better performance after adding the reconstruction loss in the auto-encoder. Minimizing the reconstruction loss can ensure that the auto-encoder is learning effective feature representation (i.e., can be decoded during the training process).

3.2.4 The Impact of Spatial Constraint. Figure 10 presents the accuracy (RMSE) as a function of γ by fixing $\alpha = 2$ and $\beta = 10$. We observe that when $\gamma = 0$ (i.e., no spatial constraint), the model had relatively low performance compared to $\gamma = 1$, which demonstrates the necessity of considering spatial dependency when dealing with the prediction of location-dependent time series data. However, when γ increases, the performance also decreases, which could due to adding excessive smooth to the prediction results.

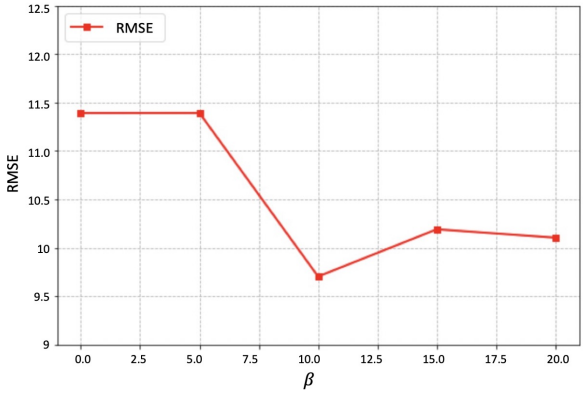


Figure 9: The RMSE as a function of β

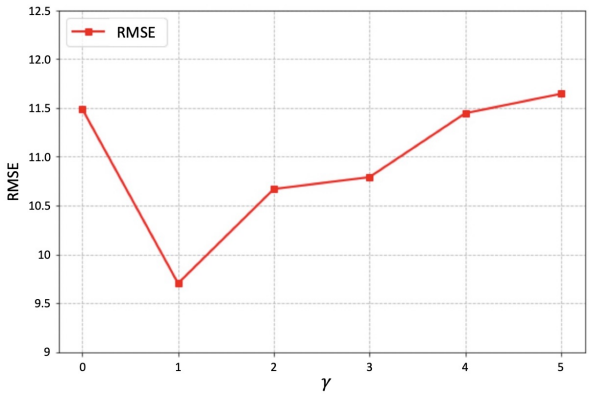


Figure 10: The RMSE as a function of γ

4 RELATED WORK

4.1 Spatiotemporal Prediction

Deep learning approaches have been developed for time series data analysis by modeling spatiotemporal relationships. Graph Convolutional Neural Network (GCN) was first introduced by [1]. [10] propose the Diffusion Convolutional Recurrent Neural Network to forecast traffic volumes by combining diffusion convolution with RNN. The graph-based RNN is flexible in modeling irregular networks like the road networks, but it is not scalable for the fine-scale predictions. [24] propose a deep multi-view network to predict taxi demand based on CNN and LSTM. [23] introduce Conv-LSTM for weather forecasting. However, the above methods are usually for forecasting problems with dense sensors, and none of them concerns the fine-scale spatial prediction.

4.2 Air Quality Prediction

The classical and most common methods for spatially predicting air quality are IDW and Kriging [21]. However, they do not explicitly consider environmental characteristics and cannot generate fine-grained predictions based on external spatially and temporally varying factors [11]. Classical dispersion models, such as Gaussian Plume models [19], usually build a function of meteorology, built environments, and emission sources based on empirical assumptions and area-specific parameters. Some required data are not easily incorporated in dispersion models, such as fine-scale traffic data. LUR models often rely on expert-selected predictors, including predictor types and spatial radii (i.e. buffers) around target areas, which cannot easily generalize to other regions. In contrast, our approach can automatically learn air quality-related predictors from multiple online datasets.

Machine learning-based models have recently become more widely used for air quality prediction. [11] propose an approach to select significant air quality-related features for prediction automatically. However, it does not take temporal effects into account. [26] propose a co-training framework with separate classifiers for spatial and temporal features. However, it fails to learn spatiotemporal effects on air quality jointly. Hsieh et al. [7] utilize an affinity graph to represent the network and apply semi-supervised learning to learn the "similarity" between locations to infer air quality values. However, graph-based approaches ignore the positional relationships between locations. More recently, [18] propose a semi-supervised learning approach to reinforce the spatiotemporal relationships of unlabeled data. However, it does not learn the effects of the built environment. In contrast, our approach employs a spatially constrained semi-supervised loss function and jointly learns spatiotemporal effects to provide accurate predictions at a high resolution.

5 CONCLUSION AND FUTURE WORK

This paper presents a novel learning architecture for fine-scale predictions of location-dependent time series data and an application of the architecture for explainable air quality prediction using publicly available and easily accessible Web data. The advantages of the presented architecture and application are that it: (1) generates explainable (air quality-related) factors from a variety of web-based open source contextual data sources, (2) considers spatial and temporal effects simultaneously in prediction, and (3) takes advantage of unlabeled data to model spatial dependencies in a semi-supervised way. The results of our model are accurate and interpretable fine-scale predictions of $PM_{2.5}$ concentrations that can be used in analyzing air quality-related health effects such as respiratory disease and asthma. Overall, the presented network architecture is generalizable and can be applied to many scientific prediction problems dealing with spatiotemporal data, sparse ground truth measurements, and requiring an understanding of the important features driving the phenomena. We plan to apply this approach to other data types such as to predict traffic and noise. We will also explore incorporating other sources of publicly-available contextual data including remote sensing observations.

REFERENCES

[1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. 2014. Spectral networks and locally connected networks on graphs. In *ICLR*.

[2] Alice Cavaliere, Federico Carotenuto, Filippo Di Gennaro, Beniamino Gioli, Giovanni Gualtieri, Francesca Martelli, Alessandro Matese, Piero Toscano, Carolina Vagnoli, and Alessandro Zaldei. 2018. Development of Low-Cost Air Quality Stations for Next Generation Monitoring Networks: Calibration and Validation of PM2.5 and PM10 Sensors. *Sensors* 18, 9 (2018), 2843.

[3] L. Chen, Y. Ho, H. Lee, H. Wu, H. Liu, H. Hsieh, Y. Huang, and S. C. Lung. 2017. An Open Framework for Participatory PM2.5 Monitoring in Smart Cities. *IEEE Access* 5 (2017), 14441–14454. https://doi.org/10.1109/ACCESS.2017.2723919

[4] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. A neural attention model for urban air quality inference: Learning the weights of monitoring stations. In *AAAI*.

[5] Shaolong Feng, Dan Gao, Fen Liao, Furong Zhou, and Xinming Wang. 2016. The health effects of ambient PM2.5 and potential mechanisms. *Ecotoxicology and environmental safety* 128 (2016), 67–74.

[6] Gerard Hoek, Rob Beelen, Kees De Hoogh, Danielle Vienneau, John Gulliver, Paul Fischer, and David Briggs. 2008. A review of land-use regression models to assess spatial variation of outdoor air pollution. *Atmospheric environment* 42, 33 (2008), 7561–7578.

[7] Hsun-Ping Hsieh, Shou-De Lin, and Yu Zheng. 2015. Inferring air quality for station location recommendation based on urban big data. In *ACM SIGKDD*. 437–446.

[8] Winnie Kam, Kalam Cheung, Nancy Daher, and Constantinos Sioutas. 2011. Particulate matter (PM) concentrations in underground and ground-level rail systems of the Los Angeles Metro. *Atmospheric Environment* 45, 8 (2011), 1506–1516.

[9] Itai Kloog, Alexandra A Chudnovsky, Allan C Just, Francesco Nordio, Petros Koutrakis, Brent A Coull, Alexei Lyapustin, Yujie Wang, and Joel Schwartz. 2014. A new hybrid spatio-temporal model for estimating daily multi-year PM2.5 concentrations across northeastern USA using high resolution aerosol optical depth data. *Atmospheric Environment* 95 (2014), 581–590.

[10] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

[11] Yijun Lin, Yao-Yi Chiang, Fan Pan, Dimitrios Stripelis, José Luis Ambite, Sandrah P Eckel, and Rima Habre. 2017. Mining public datasets for modeling intra-city PM2.5 concentrations at a fine spatial resolution. In *ACM SIGSPATIAL*. 25.

[12] Yijun Lin, Nikhit Mago, Yu Gao, Yaguang Li, Yao-Yi Chiang, Cyrus Shahabi, and José Luis Ambite. 2018. Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning. In *ACM SIGSPATIAL*. 359–368.

[13] AG Megaritis, C Fountoukis, PE Charalampidis, HAC Denier Van Der Gon, C Pilinis, and SN Pandis. 2014. Linking climate and air quality over Europe: effects of meteorology on PM 2.5 concentrations. *Atmospheric Chemistry and Physics* 14, 18 (2014), 10283–10298.

[14] DK Moore, Michael Jerrett, WJ Mack, and N Künzli. 2007. A land use regression model for predicting ambient fine particulate matter across Los Angeles, CA. *Journal of Environmental Monitoring* 9, 3 (2007), 246–252.

[15] Clare S Murray, Gina Poletti, Tatiana Kebadze, Julie Morris, Ashley Woodcock, S. L. Johnston, and Adnan Custovic. 2006. Study of modifiable risk factors for asthma exacerbations: virus infection and allergen exposure increase the risk of asthma hospital admissions in children. *Thorax* 61, 5 (2006), 376–382.

[16] E. Patterson and D. J. Eatough. 2000. Indoor/outdoor relationships for ambient PM2.5 and associated pollutants: epidemiological implications in Lindon, Utah. *J. Air Waste Manag. Assoc.* 50, 1 (2000), 103–110.

[17] David Y.H. Pui, Sheng-Chieh Chen, and Zhili Zuo. 2014. PM2.5 in China: Measurements, sources, visibility and health effects, and mitigation. *Particuology* 13 (2014), 1–26.

[18] Zhongang Qi, Tianchun Wang, Guojie Song, Weisong Hu, Xi Li, and Zhongfei Zhang. 2018. Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. *IEEE TKDE* 30, 12 (2018), 2285–2297.

[19] Sotiris Vardoulakis, Bernard EA Fisher, Koulis Pericleous, and Norbert Gonzalez-Flesca. 2003. Modelling air quality in street canyons: a review. *Atmospheric environment* 37, 2 (2003), 155–182.

[20] Dane Westerdahl, Scott Fruin, Todd Sax, Philip M Fine, and Constantinos Sioutas. 2005. Mobile platform measurements of ultrafine particles and associated pollutant concentrations on freeways and residential streets in Los Angeles. *Atmospheric Environment* 39, 20 (2005), 3597–3610.

[21] David W Wong, Lester Yuan, and Susan A Perlin. 2004. Comparison of spatial interpolation methods for the estimation of air quality data. *Journal of Exposure Science and Environmental Epidemiology* 14, 5 (2004), 404.

[22] Mingyuan Xin and Yong Wang. 2019. Research on image classification model based on deep convolution neural network. *EURASIP Journal on Image and Video Processing* 2019, 1 (2019), 40.

[23] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*. 802–810.

[24] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *AAAI*.

[25] Ruiyun Yu, Yu Yang, Leyou Yang, Guangjie Han, and Oguti Move. 2016. RAQ–a random forest approach for predicting air quality in urban sensing systems. *Sensors* 16, 1 (2016), 86.

[26] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. 2013. U-air: When urban air quality inference meets big data. In *ACM SIGKDD*. 1436–1444.