Multi-Group Encoder-Decoder Networks to Fuse Heterogeneous Data for Next-Day Air Quality Prediction

Yawen Zhang[1] , **Qin Lv**[1] , **Duanfeng Gao**[1] , **Si Shen**[1] , **Robert Dick**[2] ,
Michael Hannigan[1] and **Qi Liu**[3]
[1]University of Colorado Boulder
[2]University of Michigan
[3]Unsupervised Inc.
{yawen.zhang, qin.lv, duanfeng.gao, si.shen, michael.hannigan}@colorado.edu, dickrp@umich.edu

Abstract

Accurate next-day air quality prediction is essential to enable warning and prevention measures for cities and individuals to cope with potential air pollution, such as vehicle restriction, factory shutdown, and limiting outdoor activities. The problem is challenging because air quality is affected by a diverse set of complex factors. There has been prior work on short-term (e.g., next 6 hours) prediction, however, there is limited research on modeling local weather influences or fusing heterogeneous data for next-day air quality prediction. This paper tackles this problem through three key contributions: (1) we leverage multi-source data, especially high-frequency grid-based weather data, to model air pollutant dynamics at station-level; (2) we add convolution operators on grid weather data to capture the impacts of various weather parameters on air pollutant variations; and (3) we automatically group (cross-domain) features based on their correlations, and propose multi-group Encoder-Decoder networks (MGED-Net) to effectively fuse multiple feature groups for next-day air quality prediction. The experiments with real-world data demonstrate the improved prediction performance of MGED-Net over state-of-the-art solutions (4.2 % to 9.6 % improvement in MAE and 9.2 % to 16.4 % improvement in RMSE).

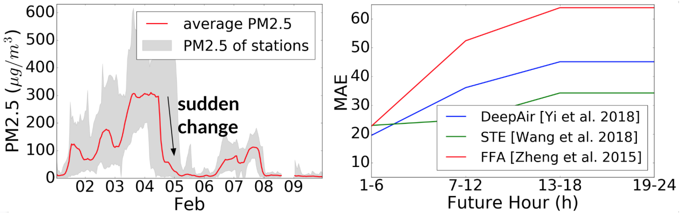


Figure 1: Challenges of air quality prediction: (Left) high spatial and temporal variability and (Right) difficulties in long-term prediction.

1 Introduction

Air pollution is a major environmental concern in urban areas. Accurate next-day air quality prediction is of particular importance for cities and individuals to cope with air pollution in the real world. With next-day air pollution warning, cities and individuals can respond in advance, for example, restricting traffic, shutting down factories, and limiting outdoor activities. Despite extensive studies, next-day air quality prediction remains a challenging problem, due to high spatio-temporal variability and difficulties in long-term prediction (Figure 1).

First, air quality is affected by complex factors and has high spatial and temporal variability. Usually, air quality monitoring stations are sparsely distributed in space. For example, there are 35 observation stations in Beijing, China, as shown in the top left subfigure in Figure 2. A recent study shows that the representation area of a station varies by both location and time, which can range from 0.25 to 16.25 km^2 and is less than 3 km^2 for most stations [Shi *et al.*, 2018]. Besides spatial variability, gradual and abrupt changes caused by various factors also induce temporal variability. Given the high spatial and temporal variability at station-level, leveraging heterogeneous data to learn those dynamics is a major challenge.

Second, predicting long-term air quality is particularly challenging. Previous studies have shown good prediction performance in the short term (e.g., next 6 hours), but the prediction error increases quickly for longer term prediction (e.g., next day), as shown in Figure 1. In particular, weather conditions play an important role in long-term prediction and it has been shown that severe haze events in Beijing are largely affected by weather conditions [Guo *et al.*, 2014]. Although previous studies have included weather forecast as input, given the high variability of weather parameters, how to effectively represent weather data and capture their impacts on air pollution dynamics requires further exploration.

When dealing with multi-source data, one key question is how to fuse them and leverage the complementary information from heterogeneous data. Currently, there exist only a few fusion models for such environmental problems. Multimodal deep learning proposes DNN-based fusion architectures to learn the joint space with multiple modalities [Ngiam *et al.*, 2011]. In the multimodal setting, for example, video which inherently includes modalities like image, motion and sound, each modality is an integrated component and multiple modalities are fused for specific tasks. Motivated by multimodal data fusion, for air quality prediction, we consider two major questions: (1) how to identify multiple modalities or feature groups from multi-source data, which are not inherently multimodal; and (2) how to conduct time series prediction with the fusion architecture, since LSTM and seq2seq models are originally designed for single feature scenarios.

To address the above-mentioned challenges, we propose multi-group Encoder-Decoder networks (MGED-Net), which fuses heterogeneous data to provide next-day air quality prediction. The main contributions of our work are:

- To capture spatial and temporal variations of air pollution at station-level, we leverage multi-source data including air pollutants, weather, road networks and elevation. A major improvement over previous studies is the utilization of high-frequency grid-based weather data from an official source.
- MGED-Net adopts a local convolution method to learn local weather impacts on air pollution dynamics. This feature representation is applied to both historical and forecast weather data and it enables more accurate air quality prediction.
- MGED-Net uses a novel structure that combines distributed fusion and sequence learning. The distributed fusion is accomplished through a grouping strategy that generates multiple (cross-domain) feature groups, and sequence learning is based on Encoder-Decoder LSTM structure.
- Extensive evaluations using real-world dataset demonstrate improved performance of MGED-net over state-of-the-art models for next-day air quality prediction.

2 Data Sources and Problem Formulation

Our study focuses on predicting next-day PM2.5 [1].

2.1 Data Sources

Based on extensive literature survey and our preliminary analysis, we have chosen the following datasets for our study.

Air Quality Data

We collected air quality data from all 35 stations in Beijing, China from January 1st, 2016 to January 31st, 2018. The stations are located in urban, suburban, near-traffic and other regions. Each station provides hourly reports of multiple air pollutants, including PM2.5, PM10, O3, NO2, CO, SO2. One major problem with the air quality dataset is missing data. The percentage of missing varies by air pollutant type (e.g., 14.7 % for PM2.5 and 28.6 % for PM10). We use linear interpolation to fill in missing data that occur within 3 hours. Continuous missing data which span longer than 3 hours are marked as NaN and not used in our study.

Grid Weather Data

Most previous studies use station-based weather data. However, there are only 17 weather monitoring stations in Beijing. Considering that some weather parameters can change significantly within a short distance, the representativeness of station-based weather data remains a serious concern. Besides, the highly nonuniform distribution of weather stations imposes further difficulties in representing weather impacts for all air quality monitoring stations.

Therefore, we choose to use grid-based weather data, which are obtained from Global Data Assimilation System (GDAS), from National Center for Environmental Prediction (NCEP) Global Forecast System (GFS) [2]. This is a data assimilation product, where multi-source observations from stations, satellites, radars, etc. are incorporated with physical atmospheric model. The spatial resolution of the grid data is 0.25 ° and it has 117 grids covering Beijing area. The selected attributes of weather data include temperature, humidity, wind speed and wind direction (further decomposed into wind_u and wind_v). While GDAS provides 3D grid data, only one layer of height 50 m (for temperature, humidity) and another layer of height 100 m (for wind) are selected. To prepare historical weather data, a temporal linear interpolation is conducted to convert the 3-hourly raw data to hourly data.

Besides obtaining historical weather data, we use a similar process to extract weather forecast data from the weather forecasts of GDAS. It should be noted that no historical weather data are included in weather forecast data. Therefore, historical and forecast weather data are separately generated by the official source and they both represent the real-world state-of-the-art accuracy.

Geo-Context Data

We include road networks and elevation data to represent geo-context of stations. The road networks data is from OpenStreetMap (OSM), and elevation is Shuttle Radar Topography Mission (SRTM) data in 80 m spatial resolution. The reason that we leverage geo-context data instead of Station ID to differentiate stations is because they explicitly describe geographic characteristics of different locations [Lin *et al.*, 2017], and can be further applied to predict air quality at non-station locations where using Station ID is infeasible.

2.2 Problem Formulation

Let s_i be the target station, where multi-source data are gathered at or around s_i. Given a time window of length T, air quality features are specified as $\boldsymbol{A} = (\boldsymbol{a}^1, \boldsymbol{a}^2, \dots, \boldsymbol{a}^T) \in \mathbb{R}^{T \times k}$, where $\boldsymbol{a}^i \in \mathbb{R}^k$ and k is the number of air pollutants observed. Historical weather features are specified as $\boldsymbol{W_h} = (\boldsymbol{w_h}^1, \boldsymbol{w_h}^2, \dots, \boldsymbol{w_h}^T) \in \mathbb{R}^{T \times h \times c}$, where $\boldsymbol{w_h}^i \in \mathbb{R}^{h \times c}$, h is the number of weather parameters observed, and c is the size of grid selected. Both historical and forecast weather data are utilized around a station. Similarly, forecast weather features are specified as $\boldsymbol{W_f} = (\boldsymbol{w_f}^{T+1}, \boldsymbol{w_f}^{T+2}, \dots, \boldsymbol{w_f}^{T+\tau}) \in \mathbb{R}^{\tau \times h \times c}$ and τ is the length of forecasting time window. The geo-context features are specified as $\boldsymbol{G} \in \mathbb{R}^m$, where m is the number of geo-context features extracted. Time features are specified as $\boldsymbol{T} \in \mathbb{R}^p$, where p is the number of time features extracted from timestamps of data points.

Problem: Given a station s_i and a target air pollutant a, historical time window T, we fuse multi-source data to predict a_{s_i} in the next τ hours, denoted as $\hat{a}_{s_i} = (\hat{a}_{s_i}^{T+1}, \hat{a}_{s_i}^{T+2}, \dots, \hat{a}_{s_i}^{T+\tau}) \in \mathbb{R}^\tau$. The purpose of the fusion model is to predict:

$$\hat{a} = \mathcal{F}(\boldsymbol{A}, \boldsymbol{W_h}, \boldsymbol{W_f}, \boldsymbol{G}, \boldsymbol{T}), \qquad (1)$$

where $\mathcal{F}$ is the fusion model.

[1] PM2.5 refers to particulate matter (PM) with a diameter of less than 2.5 micrometers, which is a major concern for air quality.

[2] https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-data-assimilation-system-gdas

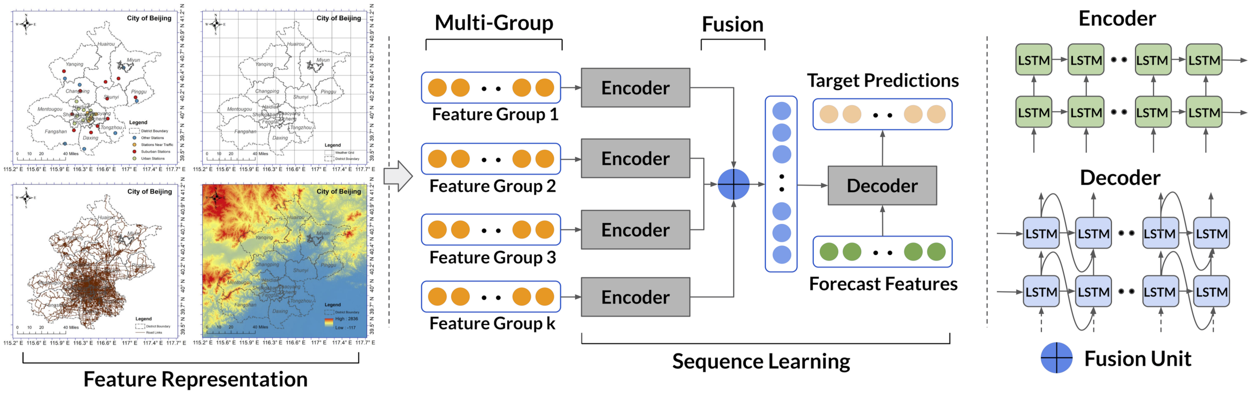


Figure 2: Architecture overview of MGED-Net: The model contains multiple Encoders and one Decoder, and both Encoder and Decoder are established using stacked LSTMs. Multiple feature groups are automatically determined and fed into different Encoders. A fusion unit is used to combine their hidden states into a joint representation, which is used as the initial state of the Decoder. After that, forecast features are fed into the Decoder, which predicts future air quality sequentially.

3 MGED-Net Model Description

We first give an overview of the proposed MGED-Net model, then describe in detail each of the key components.

3.1 Overview of MGED-Net

Figure 2 shows the overall architecture of MGED-Net. MGED-Net is intrinsically designed for multi-feature setups by first formulating the feature groups and then feeding them into the Encoder-Decoder structure. It effectively integrates complementary information from different data sources. It has three key components: feature representation, multi-group feature integration, and fusion architecture.

3.2 Feature Representation

Given various types of data obtained from each station, the first component of MGED-Net aims to extract effective features at the station level. Specifically, we investigate how to represent various data types (e.g., grid, vector) at point-based stations.

Air quality features. As mentioned earlier, each station monitors 6 types of air pollutants and reports their concentration levels hourly. All these air pollutant time series readings are directly included.

Historical and forecast weather features. The grid-based weather data characterizes the local weather around a station. With grid data, we experiment with three different options to extract weather features as shown in Figure 3. The first option is a widely used approach that extracts the nearest grid element, i.e., the grid element that a station falls in. The problem with this approach is that only limited weather information is utilized. The second and third options use the idea of "local grids", i.e., $k \times k$ grids around a station, which have better spatial coverage around the stations. While option 2 computes the mean values of weather parameters, option 3 adopts a *convolution layer* on local grids to learn the kernels for various weather parameters. The *convolution layer* is applied to capture nonuniform impacts of local weather, for example, some kernels can represent the derivatives of wind field.

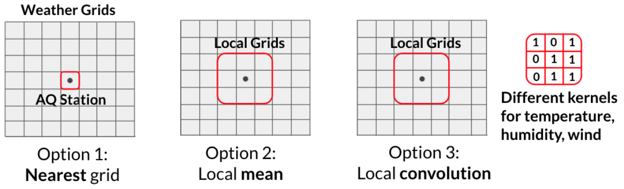


Figure 3: Different grid data representations (AQ: air quality).

Geo-context features. We convert road networks data into grid format by generating a fishnet with 10 km spatial resolution. The road density of each grid element is the number of roads passing through it. For elevation, to capture its variation, we compute the $(mean, std)$ elevation of pixels within each 10 km^2 grid element. Road density and elevation features of a station are extracted from the grid element it falls in.

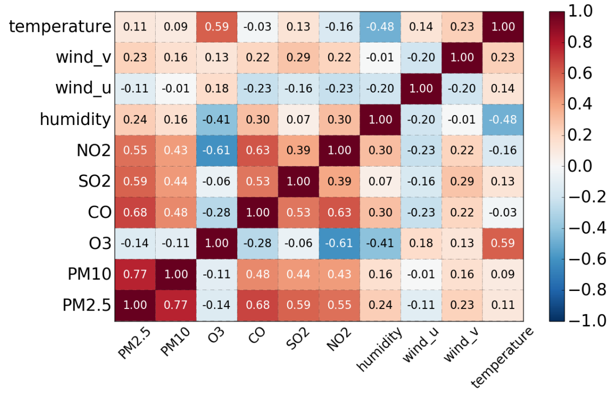
Time features. Given the timestamps of input data points, we extract 3 time features of the last timestamp: hour of day, day of week, and month.

Summary. The extracted features fall into 5 categories: air quality, historical weather, forecast weather, geo-context, and time. For prediction, these features are converted to the range of $[0, 1]$ using Min-Max Normalization, and the inverse of Min-Max Normalization is applied to recover real values.

3.3 Multi-Group Feature Integration

Given multiple features as input, the purpose of multi-group feature integration is to formulate feature groups. Intuitively, features within the same domain (e.g., air quality vs. weather) can be grouped together, which is a strategy widely used in previous studies [Yi *et al.*, 2018; Yuan *et al.*, 2018]. We refer to this strategy as "domain groups" which leverages the domains of the features for grouping.

However, this intuitive grouping strategy may not be optimal, especially given the fact that cross-domain features (i.e., features from different domain groups) may be more closely related. Figure 4 shows a correlation analysis of features from different domains. We can see that O3 (an air quality feature)



	PM2.5	PM10	O3	CO	SO2	NO2	humidity	wind_u	wind_v	temperature
temperature	0.11	0.09	0.59	-0.03	0.13	-0.16	-0.48	0.14	0.23	1.00
wind_v	0.23	0.16	0.13	0.22	0.29	0.22	-0.01	-0.20	1.00	0.23
wind_u	-0.11	-0.01	0.18	-0.23	-0.16	-0.23	-0.20	1.00	-0.20	0.14
humidity	0.24	0.16	-0.41	0.30	0.07	0.30	1.00	-0.20	-0.01	-0.48
NO2	0.55	0.43	-0.61	0.63	0.39	1.00	0.30	-0.23	0.22	-0.16
SO2	0.59	0.44	-0.06	0.53	1.00	0.39	0.07	-0.16	0.29	0.13
CO	0.68	0.48	-0.28	1.00	0.53	0.63	0.30	-0.23	0.22	-0.03
O3	-0.14	-0.11	1.00	-0.28	-0.06	-0.61	-0.41	0.18	0.13	0.59
PM10	0.77	1.00	-0.11	0.48	0.44	0.43	0.16	-0.01	0.16	0.09
PM2.5	1.00	0.77	-0.14	0.68	0.59	0.55	0.24	-0.11	0.23	0.11

Figure 4: Correlation matrix of multiple features.

is more related to temperature and humidity (two weather features). As pointed out by [Wu *et al.*, 2004], it is important to form "statistically independent" feature groups as a preliminary process for fusion, which minimize inter-group correlation and maximize intra-group correlation.

Our goal is to divide the extracted features $\{f_1, f_2, \dots, f_k\}$ into $D \in [1, k]$ feature groups. Since most features are either air quality or weather, for feature integration, we focus on air quality and weather features. We consider three different grouping strategies: *By each*, *By domain*, and *By correlation*.

By Each

With this strategy, each feature forms a group by itself, and fed into fusion model. The number of groups D equals k, which is the number of features. This strategy can be problematic when the number of features increases greatly and it relies completely on the fusion unit in learning feature interactions.

By Domain

Features can also be directly grouped by their domain categories (i.e., air quality and weather). Here, $D = s$, and s is the number of domain groups ($D = 2$ in our case). However, this grouping strategy could be harmful when non-related features are in the same group, especially when they are fed into networks with one set of parameters.

By Correlation

The intuitions this grouping strategy are: (1) intra-group correlation should be maximized, i.e., highly correlated features should be grouped together and fed into the same subnetwork; and (2) inter-group correlation should be minimized. This is a non-overlapping strategy and each feature only belongs to one group, which is different from previous approach [Yi *et al.*, 2018] where the target feature is included in all groups.

We design a clustering-based approach to group features. The correlation $c(f_p, f_q)$ between features f_p and f_q is computed as their average Pearson correlation coefficients of all samples. The inter-group correlation is measured as the average pair-wise correlation of features from group G_i and G_j, and $|G_i|$, $|G_j|$ are the number of features within each group:

$$C(G_i, G_j) = \frac{1}{|G_i||G_j|} \sum_{f_p \in G_i} \sum_{f_q \in G_j} |c(f_p, f_q)|, i, j \in D \tag{2}$$

The intra-group correlation is measured as:

$$C(G_i) = C(G_i, G_i), i \in D \tag{3}$$

where $C(G_i)$ is the average pair-wise correlations of features within group G_i. The objective function used to detect feature groups is defined as:

$$\min \frac{1}{\sum_{d=1}^{D-1} d} \sum_{i=1, j>i}^{D} [\frac{C(G_i, G_j)}{C(G_i)} + \frac{C(G_i, G_j)}{C(G_j)}], i, j \in D \tag{4}$$

The objective function aims at minimizing the average ratio of inter-group correlation to intra-group correlation. Given a small D, the computational complexity is $\mathcal{O}(D^k)$, and k is the number of features. We experiment with small D candidates $\{2, 3, 4\}$ and pick the one that minimizes the function.

3.4 Fusion Architecture

We use Encoder-Decoder LSTM model [Sutskever *et al.*, 2014] as our base network for time series prediction. The Encoder-Decoder structure can naturally separate historical and future time sequences, and LSTM is used for learning temporal dependencies in sequences.

Previously, Encoder-Decoder LSTM model is widely used in single feature scenario. Given multiple features, the question is how to fuse them into Encoder-Decoder LSTM model to get accurate prediction. Using the proposed multi-group feature integration strategy, we obtain D feature groups denoted as $\{(\boldsymbol{x}_d^1, \boldsymbol{x}_d^2, \dots, \boldsymbol{x}_d^T) \mid d = 1, 2, \dots, D\}$, where $\boldsymbol{x}_d \in \mathbb{R}^{c_d}$ and c_d is the number of features in the d-th group. We investigate three different fusion architectures. The last two approaches are based on Multiple Encoders structure and focus on *Fusion Unit* shown in Figure 2.

Feature fusion. This fusion architecture directly concatenates all the feature to form a single sequence $(\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^T)$, where $\boldsymbol{x} \in \mathbb{R}^k$ and k is the number of all features. This long vector is used as Encoder input.

Encoder fusion. Given the feature groups generated, each group $(\boldsymbol{x}_d^1, \boldsymbol{x}_d^2, \dots, \boldsymbol{x}_d^T), d \in [1, D]$ is used as the input of one Encoder, and their hidden states are further concatenated as $(\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_D)$ and used as the Decoder's initial state.

Encoder fusion + Group interactions. The group interactions are important in learning complex relationships among features. Building upon *Encoder fusion* structure, we further model group interactions based on Tensor Fusion [Zadeh *et al.*, 2017], an effective approach to learn inter-modality interactions. For our specific problem, if the target (i.e., PM2.5) is in group G_k with hidden state h_k, the interaction between group G_k and G_i is computed as the outer product between h_k and h_i:

$$h_{ki} = \begin{bmatrix} h_k \\ 1 \end{bmatrix} \otimes \begin{bmatrix} h_i \\ 1 \end{bmatrix}, i \in [1, D], i \neq k \tag{5}$$

By computing the interactions of feature group G_k with all other groups, their products are reshaped and concatenated as a sequence $(\boldsymbol{h}_{k1}, \boldsymbol{h}_{k2}, \dots, \boldsymbol{h}_{kD})$ and a fully connected (FC) layer is applied to reduce dimension. The output of FC layer is used as the Decoder's initial state.

4 Experiments and Analysis

4.1 Implementation Details

We process data from January 1st, 2016 to January 31st, 2018 and divide the samples by 8:1:1 into training, validation, and testing data. Training, validation and testing data do not overlap. The sequence lengths of Encoder and Decoder are set to 48 and 24. The size of local grids and convolution kernel are both set to 3×3. We conduct grid search to decide the optimal hyperparameter combination. We set the learning rate to 0.001, batch size to 128, and apply early stopping for model training. We use Adam to update parameters and Mean Squared Error (MSE) as loss function. We add dropout layer with rate 0.3 on Encoders and Decoder. All experiments are run on a machine with two NVIDIA GTX 1080 Ti GPUs.

4.2 Evaluation Metrics

We compute the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) for each future hour prediction, and average them into different hour ranges. Metrics unit: $\mu\text{g}/\text{m}^3$.

In the following subsections, we first evaluate the performances of different approaches for the three key components in MGED-Net, and then compare MGED-Net with other models.

4.3 Performance of Feature Representation

We compare three different options to represent grid weather data: *nearest grid*, *local mean* and *local convolution*. They are applied to both Encoder and Decoder [3].

	1-6h	**7-12h**	**13-18h**	**19-24h**
E (nearest) + D (-)	15.57	24.17	33.51	42.01
E (mean) + D (-)	15.11	21.92	31.42	39.13
E (conv) + D (-)	**15.41**	**22.09**	**28.49**	**36.17**
E (conv) + D (nearest)	15.76	18.98	22.71	28.73
E (conv) + D (mean)	16.00	19.94	22.43	26.72
E (conv) + D (conv)	**14.36**	**18.86**	**20.27**	**22.35**

Table 1: Grid data representations vs. prediction performance (MAE), E: Encoder, D: Decoder, D (-): no weather forecast

As shown in Table 1, weather forecast data we use makes significant improvement in long term prediction, especially after 6 hours. Among the three options for representing grid-based weather data, *local convolution* performs the best for 1 to 24 hours prediction, and *local mean* achieves better results than *nearest grid* in longer term. Previous studies using nearest weather station is similar to *nearest grid* approach. The results proves the effectiveness of applying *local convolution* to capture local weather impacts on air pollution variations.

4.4 Performance of Feature Grouping

We compare different feature grouping strategies on multiple features [4]. Using *By correlation* strategy, the optimal number of groups is 4 and the generated groups are: Group 1 (PM2.5, PM10, CO, SO2, NO2), Group 2 (O3, humidity, temperature), Group 3 (wind_u), Group 4 (wind_v). The geo-context, time features are incorporated as separate groups.

	1-6h	**7-12h**	**13-18h**	**19-24h**
By each	15.79	20.05	22.31	23.94
By domain	16.20	20.73	23.03	26.95
By correlation	**14.36**	**18.86**	**20.27**	**22.35**

Table 2: Feature grouping strategy vs. prediction performance (MAE)

As shown in Table 2, *By correlation* achieves the best performance for 1 to 24 hours predictions, and *By domain* performs the worst. Though *By domain* grouping is a widely used strategy, it actually ignores the cross-domain feature relationships. When non-related features are fed into one Encoder, it needs to learn both feature interactions and temporal dependencies and the burden is too heavy. In contrast, *By correlation* feeds Encoder with highly correlated features, and the results demonstrate the improvement with this grouping strategy.

4.5 Performance of Fusion Architectures

We compare three fusion architectures: *Feature fusion*, *Encoder fusion*, and *Encoder fusion + Group interactions (GI).*

	1-6h	**7-12h**	**13-18h**	**19-24h**
Feature fusion	18.07	21.02	22.99	26.17
Encoder fusion	14.36	18.86	20.27	22.35
Encoder fusion + GI	**13.44**	**18.05**	**20.95**	**21.91**

Table 3: Fusion architecture vs. prediction performance (MAE)

Table 3 shows the comparison results, *Encoder fusion* and *Encoder fusion + GI (Group interactions)* achieve much better performance than *Feature fusion*. *Feature fusion* is an intuitive way to deal with multiple features. However, it can be problematic due to the facts that: 1) feature correlations are weakly captured by the training weights of an Encoder [Ren *et al.*, 2016], and 2) the prediction performance will degrade rapidly as input length increases [Cho *et al.*, 2014]. *Encoder fusion* is an effective way to learn the joint representation of multiple Encoders. After adding inter-group interactions, the prediction performance is further improved.

4.6 Comparison of Fusion Models

Finally, we compare our MGED-Net model with five different models including baseline and state-of-the-art models. Here, we present the best result for each model by experimenting with different parameter settings.

- **Naive approach**: Using current hour to predict all future hours, no prediction model applied.
- **LSTM**: Using historical 48 hours to predict the future.
- **seq2seq**: Stacked LSTMs in both Encoder and Decoder, and historical 48 hours for future predictions.
- **GeoMAN** [Liang *et al.*, 2018]: A *Feature fusion* architecture based on Encoder-Decoder structure, and employs multi-level attentions to learn feature importances.
- **DeepAir** [Yi *et al.*, 2018]: A distributed fusion architecture that fuses multiple FusionNets based on parametric-matrix-based strategy.

[3] *By correlation* grouping strategy and *Encoder fusion* are applied.

[4] *Local convolution* and *Encoder fusion* are applied.

	1-6h		**7-12h**		**13-18h**		**19-24h**	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Naive approach	14.87	26.33	26.00	43.16	32.21	50.70	35.45	54.79
LSTM	14.17	20.91	25.88	33.83	32.67	40.23	37.03	44.08
seq2seq	14.13	21.39	23.99	32.59	30.14	38.55	33.61	41.89
DeepAir (6 hours) [Yi *et al.*, 2018]	19.66	25.81	23.53	29.82	28.80	36.26	28.03	35.12
DeepAir (48 hours) [Yi *et al.*, 2018]	19.18	25.15	23.13	29.64	25.20	31.88	28.43	35.37
GeoMAN (6 hours) [Liang *et al.*, 2018]	13.92	19.04	19.28	25.14	23.48	30.22	28.94	36.77
GeoMAN (48 hours) [Liang *et al.*, 2018]	14.03	19.10	19.42	25.06	22.95	29.31	24.23	32.14
MGED-Net (w/o Group interactions)	14.36	20.68	18.86	26.63	20.27	28.01	22.35	31.44
MGED-Net (w/ Group interactions)	**13.44**	**17.35**	**18.05**	**22.83**	**20.95**	**26.01**	**21.91**	**26.88**

Table 4: Performance comparisons of different models.

Table 4 shows the prediction performance of different models. Among all models, MGED-Net (w/ Group interactions) has the best performance for 1 to 24 hours prediction. Compared with LSTM, seq2seq has better performance in longer hours which demonstrates the effectiveness of using Encoder-Decoder LSTM as basic network. For DeepAir and GeoMAN models, in their original settings, they use historical 6 hours sequences as input. We experiment with different sequence lengths (i.e., 6 and 48 hours) for them. With longer historical sequences, GeoMAN achieves significant improvements in longer term prediction (i.e., after 12 hours), owing to learning temporal dependency with LSTM. In contrast, DeepAir has no component for sequence learning, the improvement with longer historical hours is not obvious. With longer historical sequences, GeoMAN works much better than DeepAir. We also notice that DeepAir perform worse in short term which may due to *By domain* grouping applied as well as its fusion strategy. Compared with GeoMAN, MGED-Net with Group interactions achieves 4.2 % to 9.6 % improvement in MAE as well as 9.2 % to 16.4 % improvement in RMSE. The major problem of GeoMAN is the feature interactions are not well modeled with its *Feature fusion* architecture.

5 Related Work

Air Quality Prediction

Recent progress in air quality prediction can be divided into two categories. The first category is about data source, which explores a variety of data that can be potentially leveraged to improve prediction performance. In recent studies, weather forecast data [Liang *et al.*, 2018; Yi *et al.*, 2018] and geo-context data [Lin *et al.*, 2017] are included and proved to be effective. The other category is about prediction model. Given multi-source data, how to fuse them to provide accurate and robust predictions is still a challenging problem. Previous models like Multi-Kernel Learning [Zheng *et al.*, 2015] fuse different data sources by separating them into spatial and temporal views, while more recent DNN-based models focus more on designing the fusion architecture [Liang *et al.*, 2018; Yi *et al.*, 2018] to fuse multi-source data.

Data Fusion

The purpose of data fusion is to leverage the complementary information from multi-source data [Ngiam *et al.*, 2011; Liu *et al.*, 2018]. Both traditional and DNN-based approaches have been explored [Zheng, 2015; Alam *et al.*, 2017]. Multimodal learning has been a growing trend in machine learning field and is closely related to our work. A variety of DNN-based fusion models have been proposed. There are two main strategies: early fusion and late fusion [Nojavanasghari *et al.*, 2016]. Early fusion (also called Feature Fusion) simply concatenates all the features into a long vector as model input [Liang *et al.*, 2018], while late fusion leverages information learned from each modality and fuses them with specific approaches [Ngiam *et al.*, 2011]. For late fusion, different methods have been proposed. For example, LSTM fusion concatenates all hidden states from LSTM [Long *et al.*, 2018], and DNN-based fusion combines multimodal predictions with fully connected layers [Nojavanasghari *et al.*, 2016; Li *et al.*, 2017]. For air quality prediction, an attention-based feature fusion model was proposed [Liang *et al.*, 2018], which leverages an Encoder-Decoder structure for sequence learning and multi-level attentions to learn feature importances. And a parametric-matrix-based fusion model has been proposed, which works as a form of ensemble model by learning weights for predictions from different components [Yi *et al.*, 2018]. Compared with existing fusion models for air quality prediction, MGED-Net model takes advantages of both distributed structure and sequence learning. As a result, both complex feature interactions and temporal dependencies can be learned to improve prediction accuracy.

6 Conclusions and Future Work

In this work, we propose multi-group Encoder-Decoder networks (MGED-Net) for next-day air quality prediction. MGED-Net consists of 3 key components: convolution-based grid data representation, correlation-based feature grouping, and multi-group fusion with group interactions. Experimental results on real-world dataset demonstrate the effectiveness of MGED-Net for next-day air quality prediction. The proposed model can also be potentially used in other multi-source data problems. For future work, we would like to investigate more on multi-feature relationships and fusion models which can adjust its fusion structure (i.e., different groups) with time-variant characteristics such as weather variations.

Acknowledgments

This work was supported in part by the US National Science Foundation (NSF) through grant CCF 1442971.

References

[Alam *et al.*, 2017] Furqan Alam, Rashid Mehmood, Iyad Katib, Nasser N Albogami, and Aiiad Albeshri. Data fusion and iot for smart ubiquitous environments: A survey. *IEEE Access*, 5:9533–9554, 2017.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[Guo *et al.*, 2014] Song Guo, Min Hu, Misti L. Zamora, Jianfei Peng, Dongjie Shang, Jing Zheng, Zhuofei Du, Zhijun Wu, Min Shao, Limin Zeng, Mario J. Molina, and Renyi Zhang. Elucidating severe urban haze formation in china. *Proceedings of the National Academy of Sciences*, 111(49):17373–17378, 2014.

[Li *et al.*, 2017] Xiang Li, Ling Peng, Xiaojing Yao, Shaolong Cui, Yuan Hu, Chengzeng You, and Tianhe Chi. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution*, 231:997 – 1004, 2017.

[Liang *et al.*, 2018] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *International Joint Conference on Artificial Intelligence*, pages 3428–3434, 2018.

[Lin *et al.*, 2017] Yijun Lin, Yao-Yi Chiang, Fan Pan, Dimitrios Stripelis, Jose Luis Ambite, Sandrah P. Eckel, and Rima Habre. Mining public datasets for modeling intracity pm2.5 concentrations at a fine spatial resolution. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '17, pages 25:1–25:10, New York, NY, USA, 2017. ACM.

[Liu *et al.*, 2018] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2247–2256. Association for Computational Linguistics, 2018.

[Long *et al.*, 2018] Xiang Long, Chuang Gan, Gerard Melo, Xiao Liu, Yandong Li, Fu Li, and Shilei Wen. Multimodal keyless attention fusion for video classification. In *The Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Ngiam *et al.*, 2011] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 689–696, USA, 2011. Omnipress.

[Nojavanasghari *et al.*, 2016] Behnaz Nojavanasghari, Deepak Gopinath, Jayanth Koushik, Tadas Baltrušaitis, and Louis-Philippe Morency. Deep multimodal fusion for persuasiveness prediction. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 284–288. ACM, 2016.

[Ren *et al.*, 2016] Jimmy Ren, Yongtao Hu, Yu-Wing Tai, Chuan Wang, Li Xu, Wenxiu Sun, and Qiong Yan. Look, listen and learn?a multimodal lstm for speaker identification. In *The Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Shi *et al.*, 2018] Xiaoqin Shi, Chuanfeng Zhao, Jonathan H. Jiang, Chunying Wang, Xin Yang, and Yuk L. Yung. Spatial representativeness of pm2.5 concentrations obtained using observations from network stations. *Journal of Geophysical Research: Atmospheres*, 123(6):3145–3158, 2018.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[Wu *et al.*, 2004] Yi Wu, Edward Y. Chang, Kevin Chen-Chuan Chang, and John R. Smith. Optimal multimodal fusion for multimedia data analysis. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 572–579, New York, NY, USA, 2004. ACM.

[Yi *et al.*, 2018] Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. Deep distributed fusion network for air quality prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery*, KDD '18, pages 965–973, New York, NY, USA, 2018. ACM.

[Yuan *et al.*, 2018] Zhuoning Yuan, Xun Zhou, and Tianbao Yang. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 984–992. ACM, 2018.

[Zadeh *et al.*, 2017] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*, 2017.

[Zheng *et al.*, 2015] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 2267–2276, New York, NY, USA, 2015. ACM.

[Zheng, 2015] Yu Zheng. Methodologies for cross-domain data fusion: An overview. *IEEE Transactions on Big Data*, 1(1):16–34, 2015.