東海大學資訊工程學系研究所

碩士論文

指導教授：呂芳懌

基於貝葉斯優化的神經網路識別空氣污染來源

Air pollution source identification using Neural Networks with Bayesian Optimization

研究生：何家昇

中華民國 108 年 7 月 14 日

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 何 家 昇 所提之論文

基於貝葉斯優化的神經網路識別空氣汙染來源

經本委員會審查，符合碩士學位論文標準。

學位考試委員會
召 集 人 羅濟舟 簽章

委 員 陳金鈴

楊朝棟

林世弘

指 導 教 授 吳元欽 簽章

中華民國 108 年 7 月 14 日

摘要

許多研究人員使用機器學習技術來提高其空氣污染濃度預測模型的準確性，讓人類可以事先得知準確的空氣汙染資訊，以避免直接曝露於汙染的環境中。據我們所知，目前並沒有研究對大範圍區域的汙染來源進行預測。為準確定位汙染來源，本研究建立了空氣污染來源識別系統，稱為 Air Pollution Source Identification System（APSIS），其使用 tensorflow 建立三種神經網路的分析模型來找出空氣汙染來源。APSIS 在相對較小的區域內收集數據，例如空氣污染濃度，風速和風向。並預先處理收集的數據，目的是確定污染物分佈是正確的，以防止 APSIS 受到異常值和其他不穩定因素的嚴重影響，如風向。使其能更準確地找出汙染來源。之後，使用高斯擴散模型進行污染物擴散模擬，並與實際擴散情形進行比較，確認高斯模型的準確性是否可以應用在污染來源辨識中，並比較三個神經網絡模型在空汙染來源辨識中的準確度，最後提出一種最適用於識別空氣污染源的模型。

Keywords: 空氣汙染，來源識別，Tensorflow，神經網路，高斯擴散模型

Abstract

Many researchers use machine learning techniques to enhance accuracies of their air-pollution concentration prediction models so that people can acquire accurate information in advance to avoid exposing themselves in this polluted environment. To the best of our knowledge, currently, there is no research which identifies air pollution source in a wide area. To accurately locate pollution sources, in this research, we create an air-pollution identification system, called Air Pollution Source Identification System (APSIS), which adopts tensorflow to establish three neural-network-based analytical models with which to find pollution sources. The APSIS collects environmental data, such as air pollution concentration, wind speed and wind direction, in a relatively smaller grid area. Next, collected data are tuned when necessary to prevent the APSIS from being seriously affected by outlier and other unstable factors, like wind direction. The purpose is to identify pollution distribution and then more accurately find out the sources. After that, the Gaussian diffusion model is used to simulate the diffusion of pollutants, and compared with the actual diffusion situation, to confirm whether the accuracy of the Gaussian model can be applied to the identification of pollution sources. Then compare the accuracy of three neural network models in the identification of air pollution sources, and finally propose a model that is most suitable for identifying air pollution sources.

Keywords: Air pollution, Source identification, Tensorflow, Neural network, Gaussian diffusion model

致謝

還記得大學三年級開始就經常聽到學長們討論碩士論文與研究，然而歲月如梭，須臾彈指間，終於我也已經跨越這考驗自我能力的挑戰。回想在研究所中的生活中，期間遇到許多困難與挑戰，所幸在各方的幫助下，方能讓我順利完成碩士論文。其中最感謝我的指導教授呂芳懌教授，在他悉心的帶領和指導下，逐步跨越各種難題。雖然在過程中也曾駐足不前而懈怠過，但感謝實驗室與研究所的學長姐與同學能夠給予我許多寶貴的意見，克豪、威昇、勝政、美瑜、奕珣、益群，建良，在與大家一起討論資訊技術與合作進行研究計畫的過程中獲益良多，讓我能補足論文與實驗中，一些不盡人意的地方，使我的論文更加完善。除此之外，也使我在研究所就讀的期間中，不僅學習到團隊合作的技巧，也磨練出面對問題時，能夠獨立解決難題的能力。這將是我在碩士生活中所獲得最寶貴的經驗與回憶。

Contents

List of Figures

List of Tables

I. Introduction

Air pollution has already existed since the very beginning of the first industrial revolution. It is a huge social cost of human beings due to economic activities. Today, when enjoying the prosperous development results of industrial revolution, we need to consider how to balance the environmental protection and economic development. In fact, this problem has plagued the world for a long time. Surrounding us, there are many pollution sources, such as air, water, soil and food. They will directly or indirectly endanger the health of human bodies. Among them, the impacts due to air pollution is the most serious one [1]. People who expose themselves to polluted air for a long time may inhale many pollutants, consequently causing diseases concerning the respiratory tract, cardiovascular and lungs, such as pneumonia, lung cancer, etc [2]. The impact on children and elderly is more serious. According to medical research and statistics [3], the incidence of disease on the respiratory tract of city residents is positively correlated with the degree of air pollution in this city. Many patients with lung cancer do not smoke, and there is no exhaust gas or air pollution in their working environments. Their common situations are living in cities with serious air pollution, and they often take exercises or perform activities outdoors. [4-7] indicated that air pollution has a significant positive correlation with lung cancer, pneumonia and human health. Our conclusion is that particulate matter (PM) in the air has a considerable correlation with lung cancer and respiratory diseases. PM can even transfer to the various parts of human bodies through throats and noses, affecting the health of other organs.

Recently, people have gradually applied information technology, like those of Internet of Things(IOT), cloud computing, big data and artificial intelligence, to study different topics of air pollution. Many research projects focus on this issue. Their common activities are preparing air quality sensors and relatively large monitoring stations to monitor air pollutants in our surrounding environment [8-9]. However, many monitoring stations, besides air pollutants, only gather temperature, humidity, wind speed etc., as assistant data from which it is often hard

for users to know which monitoring stations are closer to the pollution sources. Further, the positions and the number of pollution sources are identified by using collected wind speed, and air diffusion models, with which it is not easy to validate the identification accuracies and their reliabilities.

Generally, the best method to improve air pollution is to find out the pollution sources. To achieve this, in this study, we first build a small-area sensor network, and analyze the sources and diffusion of PM, including PM2.5, PM10 and PM1.0. Next, we propose an air pollution monitor system, named Air Pollution Source Identification System (APSIS), which identifies positions of air pollution sources based on the data collected in this small area. During our data collection stage, the degree of produced pollution is equal to that of household barbecue to avoid seriously affecting air quality of our own environment. Because of the small-scale field, the pollution sources are known. So the credibility of collected data and the identification of the pollution sources are accurate. We also simulate the pollutant distribution in a windy environment by Gaussian diffusion model. To check whether the Gaussian diffusion model help the APSIS to identify the pollution source or not.

The rest of this paper is organized as follows. Section 2 explores related studies of this research. Section 3 describes the APSIS's system architecture, data collection and preprocessing. Our experimental results are presented and discussed in Section 4. Section 5 concludes this paper and discuss our future studies.

II. Related Work and Background

Due to the low cost and small size, IoT devices have been gradually employed by many studies to observe air pollution. Rajesh *et al*. [10] used wireless sensors to monitor air pollution and proposed a sensor network establishment method, with which an established system can achieve energy efficiency and allow the IoT to instantly observe and evaluate health risks in

specific areas. Chen *et al*. [11] introduced an air pollution IoT system which uses a large number of low-cost sensors to collect a huge amount of data and ensure their monitoring accuracy. They predict air pollution through a background system and analyze the collected big data. In recent years, many studies have tried to predict the concentration, explore the causes and track the source of air pollution with IoT systems and machine learning techniques.

2.1 Air Diffusion Models

According to different mathematical and physical methods, air diffusion models can be divided into many types, including Gaussian diffusion model [12,13], Lagrangian Stochastic (LS) model [14,15], Computational Fluid Dynamics (CFD) model [16,17] and Community Multiscale Air Quality (CMAQ) [18,19], in which Gaussian diffusion model and CMAQ are the most widely used ones.

CMAQ [18] is a model established in a grid environment for collecting atmospheric science and air quality information with which to analyze air quality. The CMAQ has its own extended models, like coordinate models and weather simulation systems, to simulate the topography, geographical coordinates and various meteorological parameters of the real environment. It calculates pollution diffusion and accumulation, and explores the interaction between pollution sources and meteorological conditions. In addition, CMAQ uses the sparse matrix to simulate the size and location of pollution sources through the Sparse Matrix Operator Kernel Emissions (SMOKE) model. Many studies adopted this model to predict their air quality and meteorology [19].

The Gaussian diffusion model has a simple mathematical expression, which is usually applied to point source diffusion. This application is quite practical in the case of flat terrain [12]. Ma *et al*. [20] tested the performance of Gaussian, LS, and CFD model. The results show that the LS model has higher prediction accuracies than those of the other two models. The mean square error of LS model is 506.17. But the calculation time is long, longer than 24 hours.

The Gaussian diffusion model has the lowest prediction accuracies. Its mean square error is 1676.21 which is still within the acceptable level and the model's computational efficiencies are the highest among the three, taking only a few seconds. Therefore, to establish a fast and accurate air diffusion analysis model, the Gaussian diffusion model is one of the best choices.

2.2 Tensorflow

TensorFlow as an open-source machine-learning tool [21] enables large-scale machine learning in a variety of environments. It uses data flow diagrams to represent the state and value of each operational flow, and utilizes the Tensor Processing Unit (TPU) mode, which is an ASIC designed for machine learning, to support high-throughput operations. The use of TensorFlow can significantly reduce the difficulty of machine learning and the corresponding development costs, allowing developers to focus on their system optimization and training processes.

2.3 Back Propagation Neural Networks

Artificial neural networks (ANN) is a machine learning technique. It simulates human brain neurons for information analysis and decision making [22]. An ANN consists of an input layer, an output layer and k hidden layers, $k \geq 1$. Each layer has multiple artificial neurons, and each neuron has a specific weight, bias, and activation function as its parameters. A neuron uses these parameters to establish the relationship between input and output layers. If we do not adopt the activation function, the ANN is essentially a linear regression model. The Back Propagation neural networks (BPN) adopts an activation function, so the input can be nonlinearly transformed and back propagated. It is the reason why it can learn and execute more complex tasks, such as voice recognition and handwriting recognition. In a supervised learning system, we need to set the initial value of bias and weight to a neural network, and feed the system with the learning data and labels. The system will first calculate the error between the predicted value and the actual value by using its loss function, and then back propagate the error.

It utilizes the chain rule to calculate the partial derivative of the entire error value for a weight, and modify the weight and bias values by adopting the gradient descent mechanism. After several iterations, the best model can be obtained.

2.4 Convolutional Neural Networks

The architecture of convolutional neural networks (CNN) is composed of a convolutional layer, pooling layer and fully connected layer. Basically, this architecture accepts image data as its input, meaning that it can recognize images [23]. When we input an image, of course to the convolutional layer, the convolution kernel will extract image features by invoking convolution calculation, and generate multiple feature maps. Then it reduces the features through the pooling layer to effectively improve the performance of the following steps and lower the possibility of model overfitting. The most commonly used methods of pooling include maximum pooling and average pooling. At last, the image feature values are input to the fully connected layer for flattening, and the subsequent execution process is something like that of a BP neural network to train and test the underlying model.

2.5 Recurrent Neural Networks

Recurrent neural networks (RNN) are mainly used to analyze sequential data [24], such as air pollution, climate and traffic flow, etc., in each of which current data is related to its previous ones, meaning that RNN remembers the past. The characteristic and state are applied to analyze future data. However, when the RNN performs backpropagation to tune its model parameters, vanishing gradient or exploding gradient may occur. Especially if the number of layers is too many, the weights are multiplied by the derivative of the activation function, which may cause the gradient approaches 0, then the gradient vanish, or the gradient approaches infinity, causing the gradient explode. Thus, the performance of the trained model will not be as expected. So usually RNN is employed together with the functions of Long Short-Term

Memory (LSTM).

LSTM is a model that adds three control units, including input gate, output gate, and forget gate, to the RNN infrastructure. In the LSTM memory cell, the input gate determines whether it needs to write the underlying neuron's previous state to memory or not. The forget gate forgets or deletes some memory, and then delivers a certain amount of memory to the next step through its output gate. Integrating the LSTM and RNN models can simplify computational process and solve the problems of vanishing gradient and exploding gradient [25].

2.6 Hyperparameter Selection

A neural network model's performance is often related to the parameters selected. Different parameters may generate different outcomes on different learning models. Golovin *et al*. [26] utilized Google Vizier, an internal tuning system developed by Google, to optimize its parameters. It has many optimization methods, such as Bayesian optimization, Random search, and Genetic algorithm, by using each of which we can adjust parameters to find an optimal hyperparameter combination. With the combination, developers can save their time for arranging and combining different parameters. Experiments indicate that Bayesian optimization usually increases the model's predictive accuracy compared to that of a manually adjusted model. However, Google Vizier is not an open-source system. We need to pay for hosting services and tuning through Google CloudML. On the other hand, the functions of Advisor are similar to those of Google Vizier [27]. Advisor provides open source on Github, allowing users to tune the hyperparameters by using Random search, Grid search and Bayesian optimization. It also supports the framework of TensorFlow.

2.7 Bayesian Optimization

Hyperparameters such as number of hidden layers, number of neurons in a layer, learning rate, and activation functions required by neural networks will affect performance of the

resultant model. Bayesian optimization optimizes these hyperparameters based on Bayes' theorem. [28] utilize Bayesian optimization to tune AlphaGo multiple times so as to improve its win-rate from 50% to 66.5% in self-play games. It adopts Gaussian process [29] to establish the posterior probability model of the analyzed data's objective function, and then selects the best hyperparameter combination according to its posterior probability distribution for further test and iterative operations. However, Bayesian optimization has its disadvantage which is constantly sampling training data near local optimal points, hence often unable to evaluate all sample points, and then unable to achieve global optimization.

Therefore, when selecting the sampling points, it is necessary to individually calculate the means and variances of the hyperparameters. The larger the mean and variance, the more likely the hyperparameters have better performance. However, the maximum value of mean and variance may appear in different places simultaneously. A large mean indicates that the global optimal solution may be in this region. "Exploitation" shows that the means of sampling points are large. A large variance represents that this area has not been explored. Therefore, it is worth to explore since a global optimal solution may be there. "Exploration" is to sample a point which has a large variance. In Bayesian optimization, acquisition function is used to select Exploitation or Exploration to efficiently sample and select the best values for all hyperparameters.

2.8 Related Studies

Many studies [31-33] have tried to predict the concentrations of air pollution for their surroundings. Markiewicz [30] used complex mathematical models of Computer Fluid Dynamics (CFD) and air diffusion model to predict gas diffusion distribution. Ma and Jin [31] proposed Community Multiscale Air Quality (CMAQ), which combines wind speed, wind direction and gas diffusion formulas, to identify coal-fired pollution sources. A power plant in Northeastern North America is used as a target. It simulated emissions and diffusion of

concentrations to find some methods for identifying pollution sources and finding the effects of diffusion. Ma [32] evaluated seasonal and meteorological conditions that impact the generation of ozone in lake regions with CMAQ, and an ozone generation estimation model was established. Its prediction accuracy was assessed by comparing its prediction results with actually observed values. Ma *et al*. [33] proposed the Gaussian-RBF network which combines the radial-basis function neural network (RBF network) and Gaussian model. They first establish a Gaussian model to analyze the diffusion of air pollution, and then invoke the RBF network to reversely derive the source of air pollution according to the diffusion and pollutant distribution of the pollutants. The results show that this model can effectively find the source of air pollution with the given air pollution diffusion and distribution.

Keresztes and Rapo [34] analyzed air pollution and meteorological data of the Ciuc basin from 2012 to 2013 by using IBM SPSS statistics. Authors tried to find out the correlation between temperature, illumination, atmospheric conditions, terrain, traffic, air pollution, and the possible causes and sources of pollution by using an air diffusion model which employs a gas diffusion mathematical formula to calculate its pollutant diffusion. The IBM SPSS statistical model utilizes historical statistics for concentration prediction. However, this model is one without learning functions. It is also hard for it to adapt to an unstable atmosphere, thus causing many unknown mutations and uncertainties.

Shaban *et al*. [35] analyzed temperature, humidity, wind speed, etc., to compare the effectiveness of SVM, M5P model trees and ANN in the prediction of air pollutant concentrations, including O_3, NO_2 and SO_2. The comparison results show that the prediction on M5P is the best. Contreras and Ferri [36] mentioned that wind is one of the most important factors affecting the spread of air pollutants. Basically, without wind speed and wind direction, a model may easily overfit the corresponding pollutant concentration analysis. Therefore, it cannot apply to real world prediction, particularly when wind direction and speed change irregularly.

Also, in the past, many studies predicted the sites with the highest concentration of pollutants as the pollution sources. When wind is weak, the area with the highest concentration of pollutants perhaps is one of the pollution sources. But when wind is strong, due to wind diffusion, it is possible that air pollutant concentrations of the downwind areas a little far from a pollution source may be higher than those of the downwind near the source. We call it strong wind effect. Most researchers mentioned that wind speed and wind direction will be key parameters of their future work since it is hard for these studies to collect these types of information simultaneously throughout the concerned area. Actually, wind speed and wind direction affect air diffusion greatly, and also increase the complexity of pollution-source prediction. If the eigenvalue is excluded, the model is often overfitted.

Contreras and Ferri [37] predicted PM2.5 concentrations by adopting wind speed, wind direction, rainfall, temperature and humidity. They used wind speed, wind direction and spatial interpolation to calculate the diffusion of PM2.5, and then predict the concentrations of PM2.5 by using random forest. The conclusion is that wind speed and wind direction are two of the most important factors affecting the diffusion and prediction accuracies of PM2.5.

Kurt and Oktay [38] proposed a geographic forecasting model by using the combination of ANN and geographic models (GFM_NN). Authors collected data from 10 local monitoring stations to train their prediction model, with which to predict the concentration of PM10, SO_2 and CO for the next three days. In [39], a deep convolutional neural network was employed to analyze the pictures showing the PM2.5 concentrations. Photos of PM2.5 concentrations are classified into different categories to train the neural networks, so that the machine can analyze current air quality through those currently collected photos. In [40], the mean and variation of the meteorological data and pollutant concentrations used to characterize data are provided by the air monitoring stations in Taiwan, and LSTM is employed to predict the air pollution concentrations for the next 72 hours.

Bahari *et al*. [41] utilized traffic, wind speed, wind direction, temperature, humidity and

other data collected during the time period from 2012 to 2013 to predict the PM2.5 concentrations for the next 3 days. Lary *et al*. [42] predicted the global space-time variation of PM2.5, using the data gathered from 8329 monitoring stations in 55 countries during the time period from 1997 to 2014. Feng *et al*. [43] integrated wavelet transformation and an ANN model to predict PM2.5 concentrations for the next two days. Zheng *et al*. [44] utilized neural networks and linear regression models to simulate spatial and temporal air quality distribution and predict air quality for 43 cities in China. The predicted data are hourly updated for the next 48 hours.

Ong *et al*. [45] used a deep recurrent neural network model and the data provided by the National Institute for Environmental Studies (NIES) in Japan to predict future concentrations of PM2.5. The features adopted include wind speed, wind direction, temperature, humidity and PM2.5 concentrations. After learning, the performance of the proposed model is more superior than that of the NIES's system.

Up to present, only a few studies employed neural networks to predict air pollution sources. The reason is that reliable air pollution data and sources are difficult to obtain. With supervised learning, we need to give the machine a label to show that this is a pollution source so that the developed system can know the features of these sources. Even with semi-supervised or unsupervised learning, and the proposed models are well-trained, in our real environment, there are no actual data labels with which we can know the accuracy and reliability of the prediction results. So we need to collect reliable data and develop some methods to overcome these problems.

III. The Architecture of the APSIS

Figure1 shows the processing flow of the APSIS, in which Air Boxes are deployed to build an Internet of Things (IoT) field for the measurement of air pollutant concentration and collection of wind data. After that, we preprocess the collected data and input the data into an air diffusion model. Also, CNN, BPN and RNN neural network models developed on

Tenserflow are employed to predict the sources of air pollution. Next, Bayesian Optimization is applied to adjust the hyperparameters of the three neural network models, including the number of hidden layers, number of neurons, activation functions, learning rates and optimizers, so as to comprehend which model performs the best.

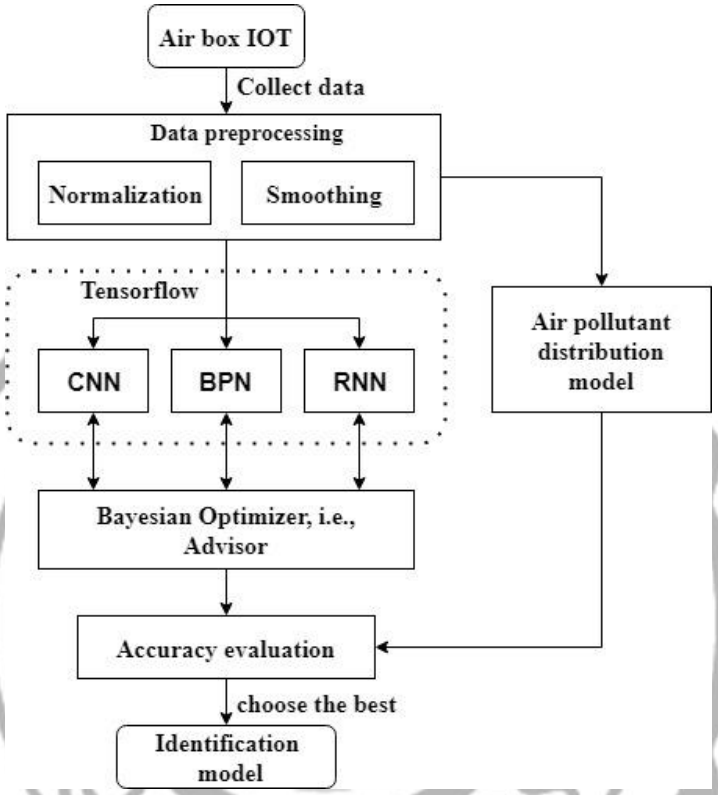


Figure 1. The processing flow of the APSIS.

3.1 Establishing IoT-based Air Boxes

In order to confirm the reliability of the trained model, when training a neural network model, it is necessary to know its pollutant concentration and each data is collected directly from a pollution source without employing any diffusion model. Basically, high-density IoT nodes are required to identify the labels of the pollution sources. Figure2 illustrates the IoT field which, as a square area of 25 m long and 25m width, is divided into 25 grids of equal size. A total of nine air boxes, also called IoT nodes, are installed. Each contains a dust sensor, a wind-speed sensor and a wind-direction sensor all connecting to an Arduino, as shown in Figure3.

Four air boxes are placed at the four corners of the grid, and the remaining five as shown in Figure2 are put in the middle of the IoT field. One or two pollution sources are randomly placed at one or two of the middle five grids. Owing to small fields, the diffusion of pollutant can be more accurately measured.

The purpose of this study is to find the location of the pollution sources through machine learning technique, rather than relying entirely on deploying high-density sensors. We placed sensors in a downwind of pollution, trying to find the highest pollution concentration. However, in a real situation, wind direction often changes greatly. Sometimes, the angle of instantaneous wind direction may change 180 degrees. Further, due to the principle of heat rise [46], the response time of sensors and the instability of the smoke source, inconsistencies may occur. For example, as experiments are performed at different time points, and the location of the pollution source and the wind direction are individually the same, the air pollution concentration collected by the same IoT node placed at the same location is different. Generally, sensors placed at the downwind near air pollution source ought to be able to detect high pollution. But sometimes owing to unstable smoke source and/or the rising of hot air, a sensor, e.g., N, on the contrary detect pollution concentrations which are lower than those detected by the sensors in N's downwind. It is also possible that when wind direction changes, and pollutant distribution is unstable, the relationship between wind direction and pollutant distribution is not completely consistent. However, such problems are inevitable, particularly when we detect pollution in an open field with a real-time manner.

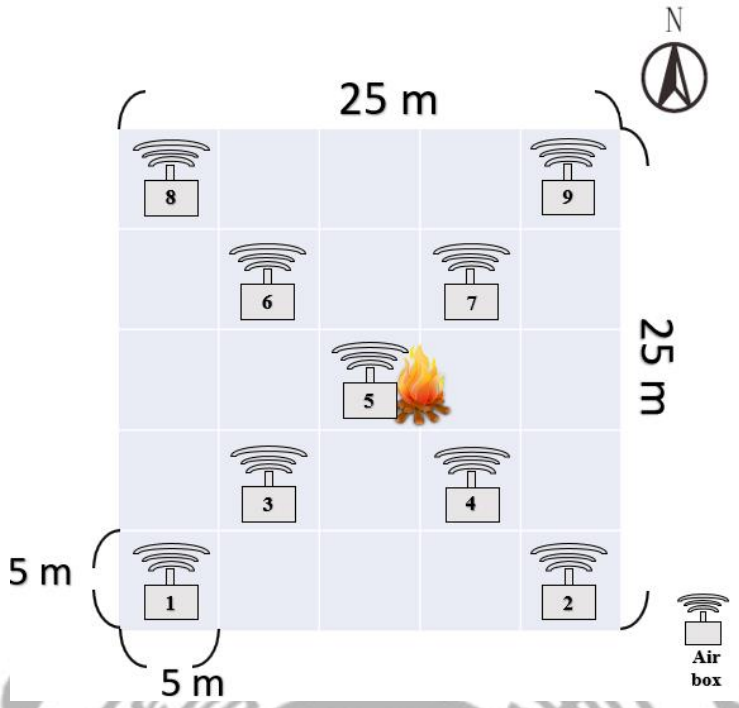


Figure 2. The layout of IoT nodes.

Figure 3. An air box, i.e., an IoT node, consists of a dust sensor, a wind-speed sensor and a wind-direction sensor all connecting to an Arduino.

3.2 Data Collection and Preprocessing

This section will discuss how data are collected and preprocessed.

3.2.1 Data Collection

The IoT field is established in Taichung City, Taiwan, gathering air pollution concentrations produced by our pollution sources from October 2018 to March 2019. The number of pollution sources may be one or two. Each of the 9 IoT nodes collects data every ten seconds, a total of 11,934 data records has been acquired. Among them, 8833 are one-pollution-source records. The remaining 3101 are dual-source records. Each data record contains data collected by the 9 air boxes, i.e., a total of 107406 (=11934*9) records of sensor data, called sensor records, are gathered. Each sensor record consists of 8 variables, including wind speed, wind direction, PM1.0, PM2.5, PM10, position of the node, time and the location of pollution source. In order to increase data diversity, the experiments were conducted across three seasons from autumn to spring.

In Taiwan, wind directions vary in different seasons. We collect data one time or two times a week. Each time lasts about one hour. During data collection, the locations of pollution sources are different. Since wind direction and strength changes drastically, it is hard for us to find time-series correlation among the pollution concentrations. So we do not adopt time data as one of machine-learning features. In addition, due to fixed positions of nodes, positions are also not the APSIS's feature. Of course, data collected by IoT nodes are orderly input to train the APSIS. Basically, we only use the remaining seven variables listed in Table1 as our pollution-source-identification features. In supervised learning, we give the machine a label defined as the exact location of pollution source, allowing the machine to define the relationship between the features and the label. The purpose is to identify whether or not there a pollution source is placed in the grid where a node locates, so as to learn and calculate the identification error for system adjustment.

Table 1. Features of the APSIS.

Variables (lag 10 sec)	Unit
I. Pollutant concentrations	
(1) PM 1.0	$\mu g/m^3$
(2) PM 2.5	$\mu g/m^3$
(3) PM 10	$\mu g/m^3$
II. Meteorological features	
(1) Wind direction	Radians
(2) Wind speed	m/sec
III. Others	
(1) Positions of the IoT nodes	1 to 9
(2) Location of a pollution source	T or F

3.2.2 Data Preprocessing

As mentioned above, some factors, e.g., abnormal data values, may negatively affect the identification accuracy of the APSIS. To reduce the affection, smoothing and normalizing data are popularly utilized. So we need to gather statistical data and then process the statistical data with the following procedure. First, we classify the data collected by a node based on wind direction and location of pollution source. Because the concentration levels of PM1.0, PM2.5, and PM10 in each data record of each IoT node are different, we consider that they are three categories in each node. Therefore, 35,802 (= 11,934*3) category records are then generated. Each category has 11,934 category records, called a category group. In other words, we have a total of three category groups, named c-group 1.0, c-group 2.5 and c-group 10.

For each category group, we first sort the 11,934 category records on their wind directions, second on location of pollution source and third on concentration scales for a node, meaning that different nodes are processed separately. After that, those category records of the same wind direction and same location of pollution source are clustered, e.g., for c-group 1.0, given a node, e.g., N, a wind direction WD and a location of pollution source PS. Due to sorting, those records,

some of N's 11,934 PM1.0 category records, that meet WD and PS will be retrieved, e.g., a total of M category records, M ≤ 11,934. PM 1.0's concentrations of which are between 0 and100 ug/m^3, belong to PM1.0-cluster 0 (WD, PS), those from 100 to 200 ug/m^3 are classified into PM1.0-cluster 1 (WD, PS), and so on. The scale difference from the max scale to the min scale of a cluster is 100 ug/m^3.

For each (WD, PS) pair, the category records in the two clusters with the largest amount of category records are considered as the normal data of (WD, PS). But, if the two clusters are not direct neighbors, we consider that the cluster with the largest amount of category records are normal data of this (WD, PS). The rest records of (WD, PS) are abnormal data which is then substituted by invoking the interpolation method for data smoothing. The substitution process is that we first average all normal data, i.e., category records, on their concentration scales, e.g., A is the average. For each abnormal category record of (WD, PS), e.g., D, a random value R, ranging between 0 and one-fifth of A, is added to A, resulting in a new random value, i.e., A + R, which is then substituted for D. After that, the other (WD, PS)s are processed by using the same method until all category records in c-group 1.0 are processed. Then we repeat the same procedure to process category records in c-group 2.5 and c-group 10 individually.

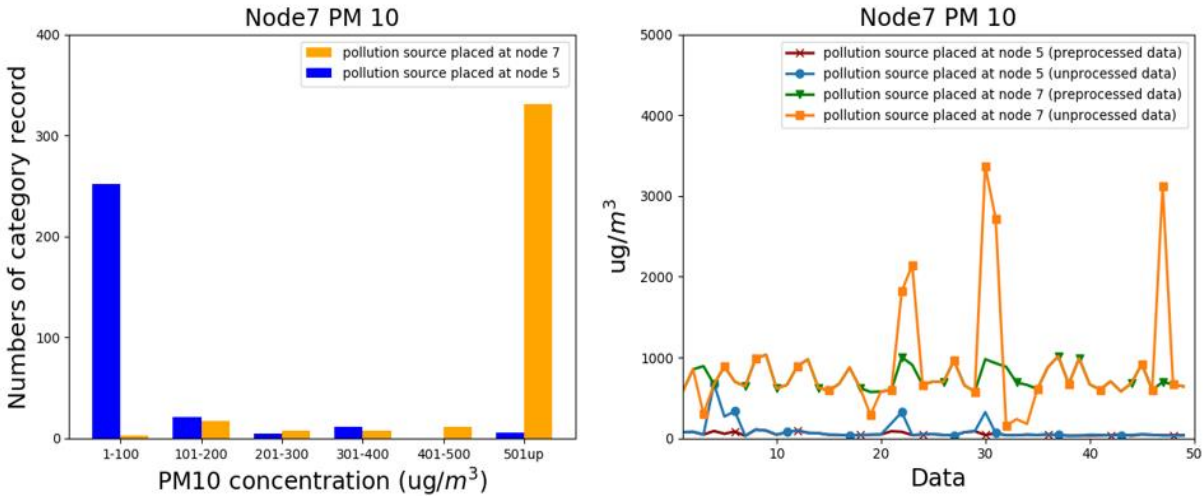
Take the PM10 measured at node 7 as an example. Figure 3 shows the statistics data of node7 when the wind blows from northeast to southwest and there is only one pollution source located at node 5 or 7. Figures 3a shows the numbers of category records of PM10 on different concentration scales for node7, whereas Figures 3b illustrates 50 category records of the air pollution concentrations before and after our data smoothing. We can see that after data smoothing, the fluctuation of PM values is mitigated. Table 2 lists the numbers of smoothed category records. The total smoothing rate of one pollution source (two pollution sources) is 9.5% (12%). Among them, since nodes 1, 2, 8, and 9 are placed at the four corners, the impact on them by pollution sources is small, hence conducting less percentage of smoothed data.

In addition, in Taiwan, winter winds are mainly from the northeast. The time period from

autumn to winter is monsoon transition time. Southerly winds may come during the transition time periods from winter to spring. In fact, wind direction is mainly affected by seasons, sea-land breeze and terrain of the region. Therefore, as a whole, node 3 located at the northeast monsoon downwind is the one highly suspected as a pollution source. However, the data collected at node3 varies seriously, consequently conducting many inaccurate data.

Summary

1. A data record consists of the data collected by 9 IoT nodes, i.e., (sensor-record 1, sensor-record 2, … sensor-record 9). There are totally 11,934 data records.
2. A sensor record comprises the data collected by an IoT node, i.e., wind speed (WS), wind direction (WD), PM1.0, PM2.5, PM10, position of node (PN), time and location of pollution source (PS). There are totally 107,406 (=11,934*9) sensor records.
3. For a category, e.g., PM1.0, there are 107,406 category records, named c-group 1.0. A category record is consisted of (WS, WD, PM1.0, PN, PS), c-group 2.5 and c-group 10 have the similar category record content, i.e., (WS, WD, PM2.5, PN, PS) and (WS, WD, PM10, PN, PS) and | c-group 1.0 | = | c-group 2.5 | = | c-group 10 | =107,406.



(a) Statistics of PM10 on node7

(b)Statistics before and after data smoothing

Figure 4. Statistical data collected at node7 for PM10 when the wind direction is 45 radians and pollution source is individually placed at node5 or node7.

Table 2. Total numbers of smoothed category records on each IoT nodes. Each IoT node collects 11934 sensor records in which there are 35802 (26499 + 9303) PM x category records, x= 1.0, 2.5 or 10.

IoT node	Number of smoothed category records	
	One source (8833*3 = 26499)	Dual sources (3101*3 = 9303)
1	1,697 (6.4%)	996 (10.7%)
2	1,488 (5.6%)	582 (6.2%)
3	4,884 (18.4%)	1,794 (19.2%)
4	2,958 (11.1%)	1,613 (17.3%)
5	4,278 (16.1%)	1,599 (17.1%)
6	3,078 (11.6%)	1,455 (15.6%)
7	2,578 (9.7%)	878 (9.4%)
8	1,011 (3.8%)	706 (7.5%)
9	851 (3.2%)	481 (5.1%)
TOTAL	22,823 (9.5%)	10,104 (12%)

After data smoothing, the Min-Max Normalization is applied to project different scales of feature values into the range between 0 and 1, making the data easier to learn and thereby improving the performance and training speed of the APSIS model. For each c-group, the i^{th} normalized data denoted by N_i is defined as

$$N_i = \frac{x_i - \min\limits_{1 \le j \le n} \{x_j\}}{\max\limits_{1 \le k \le n} \{x_k\} - \min\limits_{1 \le j \le n} \{x_j\}} \quad (1)$$

where x_i is the collected data of a category record, $x_i \in \{x_1,...,x_n\}$, n=11,934, for PM x, x=1.0 2.5 or 10, and $\min\{x_j\}$ ($\max\{x_k\}$) is the minimum (maximum) value of x, $1 \le j, k \le n$.

3.3 The Architecture of Analysis Model

This section will describe our air diffusion model and neural network model.

3.3.1 Air Diffusion Model

Our air diffusion model is developed based on the Gaussian diffusion formula shown in Eq. 2 to simulate the diffusion of pollutants. Let C_{gau} be an integrated parameter calculated as

$$C_{\text{gau}}(D_i, A_i) = \frac{1}{2\pi u \sigma_y \sigma_z} \left\{ \exp\left(\frac{-(D_z - h)^2}{2\sigma_z^2} \right) + \exp\left(\frac{-(D_z + h)^2}{2\sigma_z^2} \right) \right\} \left\{ \exp\left(\frac{-(D_y)^2}{2\sigma_y^2} \right) \right\} \quad [20] \tag{2}$$

where D_i represents the positional parameters (including downwind direction D_x, the horizontal distance perpendicular to the downwind direction D_y and D_z is the vertical distance), A_i is the atmospheric parameters (such as wind direction, wind speed (u in m/s) and atmospheric stability, etc.), h is the height of the pollution source (m), and σ_y and σ_z are, respectively, the statistical standard deviations of the horizontal and vertical dimensions in the calculation of plume where σ_y and σ_z are affected by atmospheric stability.

In the simulation of pollutant diffusion, the wind speed, wind direction and pollutant concentrations are collected. Temperature and humidity are acquired from the Meteorological Bureaus near our IoT field. The other special parameters are obtained by referring to the data of Prairie Grass emission experiment [47], e.g., wind friction velocity and atmospheric stability parameter.

At last, we simulated the diffusion of pollutants, and compared the simulation results with the data we actually observed to determine whether or not we can accurately simulate the diffusion of pollutants through Gaussian diffusion model given a small number of parameters so as to provide correct data for machine learning.

3.3.2 Neural Network Model

In this study, Python is utilized to implement the three neural network models, i.e., BPN, CNN, and RNN, on tensorflow. The data represented by array shown in Figure 5 are the input data of BPN and RNN. Pictures input to CNN as shown in Figure 6 are organized as a picture matrix based on the location distribution of the IoT nodes in the IoT field, i.e., node N's feature data in the matrix is the same as the position of the node in the IoT field. Namely, the data that CNN analyzes are a three-dimensional matrix of 5x5x5, of which the first 5x5 is the number of the flat grid in the IoT field. Each feature is expressed by a channel. Since there are a total of 5 channels, i.e., 5 features, including PM1.0, PM2.5, PM10, wind direction and wind speed. The value of 0 is given to the position of the matrix when the position provides no IoT nodes. The purposes are finding the correlation between the locations of these IoT nodes and the pollution concentrations, and improving the capability of machine in extracting and recognizing picture features.

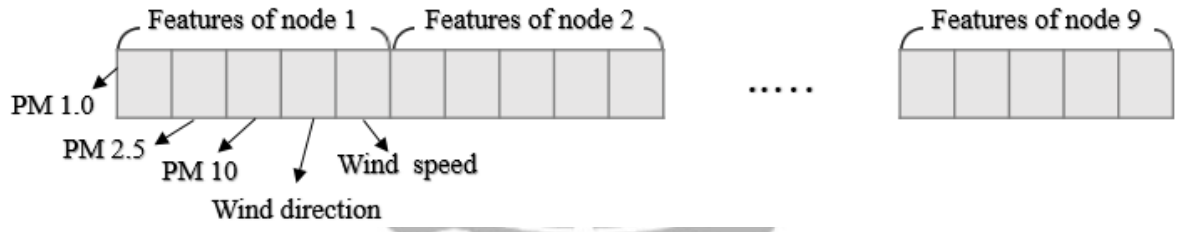


Figure 5. The data structures utilized as the inputs of BPN and RNN. Each node, e.g., node *i* contain 5 features, including PM1.0, PM2.5, PM10, wind direction and wind speed.

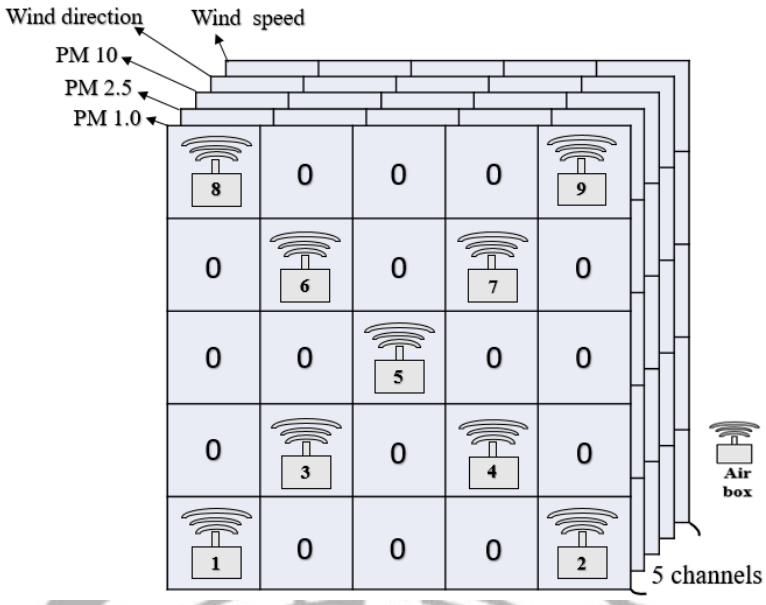


Figure 6. The data structure is utilized as the inputs of CNN.

The process of model training is as follows. (1) To reduce the possibility of overfitting, the *k*-fold cross-validation [48] method is applied to divide the single-source data into 10 sub-samples, with which to train and test the three neural network models; (2) Manually adjust the parameters to find the best identification accuracy; (3) The Advisor for Bayesian optimization is employed to recommend the best hyperparameters; (4) Compare the accuracy between manual tuning and automatic tuning, and then select the best parameters; (5) Propose the best model of BPN, CNN and RNN; (6) At last, the data of two pollution sources are input to the trained models for testing. After this, the established model is applied to analyze the accuracy of identifying multiple pollution sources. Consequently, the proposed model is the most suitable for the detection of air pollution sources.

During the air pollution-source identification stage, the data gathered by the four nodes at the corners, including nodes 1, 2, 8 and 9, are only used as the feature, without being the target pollution-source identification. Because our pollution sources have never been placed at the four points, their labels are always False. If they are added as a part of the training data, the machine will possibly identify all the IoT nodes as non-polluting sources directly to increase its

identification accuracy because most IoT nodes are denoted by false. But this is a false accuracy. In reality, it is difficult to identify the number of pollution sources from air-quality data collected by air quality monitoring stations since these data often are not accompanied with locations of pollution sources. Basically, we like that the machine can autonomously identify all pollution sources, rather than wishing a model to identify a specific number of pollution sources decided by users.

IV. Results and Evaluation

The specifications of our simulation environment are shown in Table 3. In this study, three experiments are performed. The first analyzes the diffusion of pollution concentrations. The second measures the APSIS's identification accuracy in a real environment. The third experiment evaluates identification accuracy given one pollution source or two pollution sources.

The accuracy of a model is defined as.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \tag{3}$$

where TP standing for true positives is defined as that it is true and the APSIS identification is also true, TN representing true negatives is defined as that it is false, and the APSIS identification is also false, FN meaning false negatives is defined as that it is true, but the APSIS identification is false, and FP standing for false positives is defined as that it is false, but the APSIS identification is true.Due to some level of erroneous learning of the APSIS, we need to improve its TP. The model is also verified by recall which is defined by Eq.4.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{4}$$

Through recall, we can clearly know the proportion of TP. The higher the value of recall, the higher the performance of the model.

Table 3 Specifications of our simulation environment.

CPU	Intel Core i9-9900 HK (eight cores)
Graphic card	NVIDIA GeForce RTX 2070, 8GB GDDR6
RAM	32 GB
Operating system	Linux (Ubuntu 16.04)
The version of python	2.7
The version of tensorflow	Gpu 1.8.0

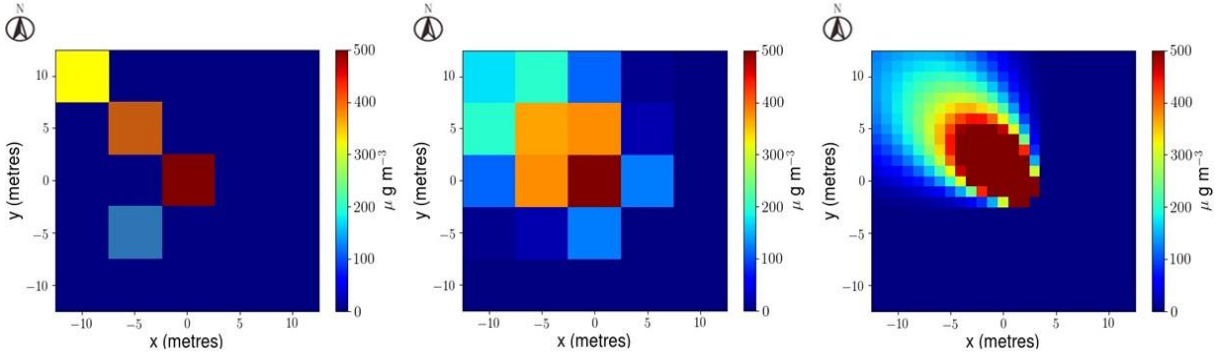
4.1 Simulated and Observed Values of Pollution Concentrations

In the first experiment, we analyzed pollutant diffusion through the Gaussian diffusion model and compared the simulation results with the actually measured concentration of pollutants. On a visualized image, the concentration of a node represents the concentration of this grid where the node is placed. We also try to adjust the size of a grid, i.e., 1 meter(m) and 5meter(m), to show the results of the Gaussian model. The simulation results are shown in Figure 7, in which the wind speed is slow and the concentration detected by the node at the pollution source is the highest. In this situation, the Gaussian diffusion model can roughly simulate a similar pollutant diffusion. But there is still a significant difference between them.

Because in a portion of the simulation results, the node, e.g., N, with the highest concentration is not always a pollution source, and we apply the Gaussian diffusion model to point out pollutant emitters given the location of N and its current wind direction. With the Gaussian diffusion model, the node with the highest concentration is considered as the pollution source. When we input N as a pollution source into the Gaussian diffusion model, even though it is not a pollution source, the results are shown in Figures 8 and 9. Since the Gaussian diffusion

model does not provide the reverse function of its calculation and learning, the simulation results and the actual results demonstrate some amount of difference.

When the wind speed is higher and the unit grid is 5m x 5m, as shown in Figures 8b and 9b, the Gaussian model is hard to simulate the pollution diffusion. But if we adjust the grid unit to 1m x 1m, as shown in Figures 8c and 9c, we can simulate the diffusion which is better than those illustrated in Figures 8b and 9b. But the results are still not very accurate. Of course, the results shown in Figure 7c are more accurately simulated compared to those of Figures 8c and 9c. We can find that when the wind speed is fast or the grid unit is large, the simulation results are very poor. However, when the wind speed is slow and the grid unit is small, the simulation results can more accurately approximate to the actual pollution diffusion. It means that in an unstable and small environment, e.g., with a fast wind speeds, the Gaussian diffusion model cannot effectively simulate the real environmental conditions due to the rapid movement of pollutants. If we can expand the range of our IoT field and gather more accurate parameters as the inputs of this model, it may effectively improve the performance of the diffusion model.

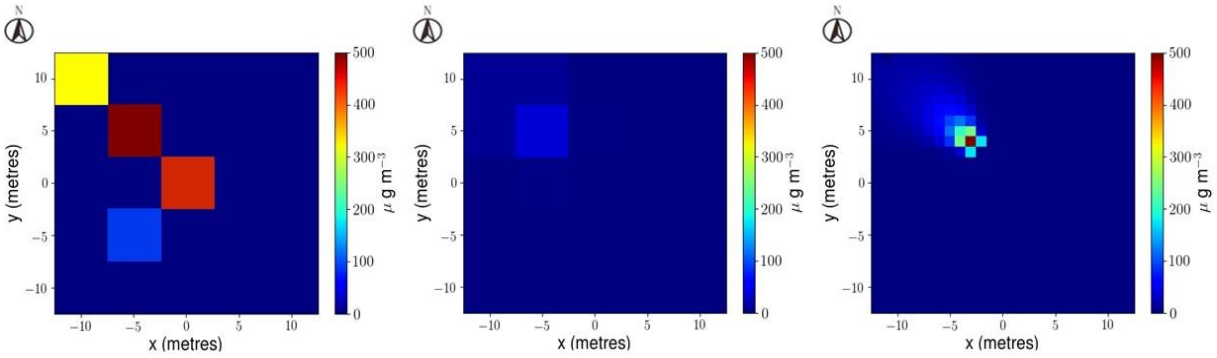


(a) Detection results of IoT nodes

(b) Simulation results of Gaussian diffusion model (size of a grid:5m x 5m)

(c) Simulation results of Gaussian diffusion model (size of a grid:1m x 1m)

Figure 7. Pollution source is placed at node5 (central grid of this IoT field, Dark red in color in Figure 7a), the concentration of PM10 is 710 μg/m^3, wind speed is 0.1m/sec and wind direction is 135 radians.

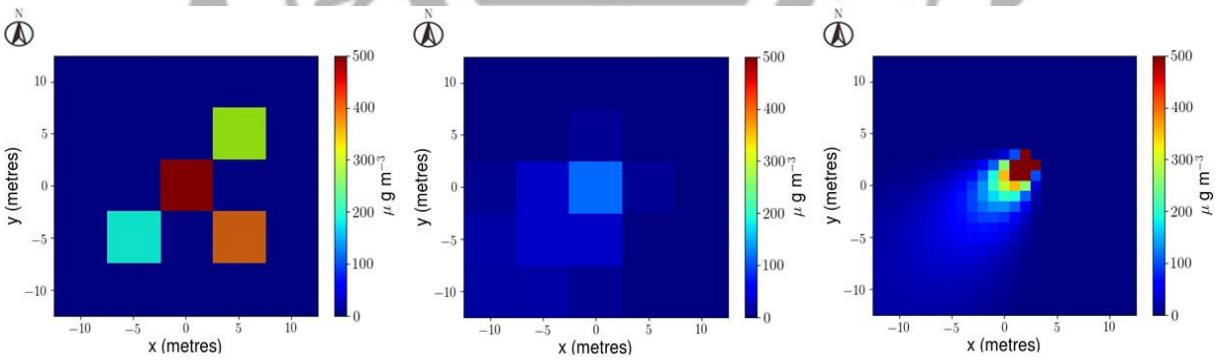


(a) Detection results of IoT nodes

(b) Simulation results of Gaussian diffusion model (size of a grid:5m x 5m)

(c) Simulation results of Gaussian diffusion model (size of a grid:1m x 1m)

Figure 8. Pollution source is placed at node5 (central grid of this IoT field, Orange in color in Figure 8a), the concentration of PM10 is 652 μg/m^3, wind speed is 2.8 m/sec and wind direction is 135 radians.



(a) Detection results of IoT nodes

(b) Simulation results of Gaussian diffusion model (size of a grid:5m x 5m)

(c) Simulation results of Gaussian diffusion model (size of a grid:1m x 1m)

Figure 9. Pollution source is placed at node 7(central grid of this IoT field, Green in color in Figure 9a), the concentration of PM10 is 611 μg/m^3, wind speed is 1.2 m/sec and wind direction is 45 radians.

4.2 Comparison of Data Preprocessing and Tuning

The second experiment is conducted in a real environment. Basically the identification accuracy is affected by environmental parameters, measurement accuracy of IoT equipment, and data inaccuracy. Therefore, we preprocess the data and compare the identification results before and after the data preprocessing. After that, parameters of the three neural network models are optimized by using the Bayesian optimization approach. Our experimental results are shown in Figures 10 and 11. It is clear that the accuracies of BPN and CNN after data preprocessing (i.e., smoothing) as shown in Figure 10 are not significantly higher than those before data preprocessing. Of course, RNN's accuracies have been enhanced obviously. But as shown in Figure 11, after Bayesian optimization, the accuracies of BPN, CNN and RNN are significantly improved, indicating that Bayesian optimization can positively tune a model's hyperparameters. Table 4 summarizes several important indicators of the two figures, where steps represent before data preprocessing, after data preprocessing and after Bayesian optimization, respectively.

Before preprocessing, the accuracies of BPN, CNN and RNN are 79.8%, 83.7% and 68.3%, respectively and the recalls of the three are 0.57, 0.64 and 0.36, respectively. Among them, RNN underperforms the other two. After data preprocessing, the accuracy of BPN reaches 87.6%, increasing 7.8%, and the recall is 0.61 after 140th epoch. The accuracy of CNN after 90th epoch is 86.5%, increasing only 2.8%, and the recall is 0.65. RNN demonstrates the most significant improvement after data preprocessing. The resulting accuracy after 106th epoch is increased by 13.4% to 81.7%, and the recall is 0.55. But its accuracy is still the worst.

Before data preprocessing, due to some inaccurate data, the accuracies of these three models are low. However, after data pre-processing, the identification accuracies of the three models have increased, and the epoch (time) of training is shortened, except that of BPN which is from 111 to 140. Basically, the effectiveness of neural network training after data pre-processing is significant. However, the recall improvements on BPN and CNN are not as

expected. From the recall, we can realize that a model attempts to learn how to find false so as to enhance its identification accuracy since most of the IoT nodes are not pollution sources, i.e., false. It is also the reason why the improvement on finding the label, i.e., identifying the pollution source, is not significant.

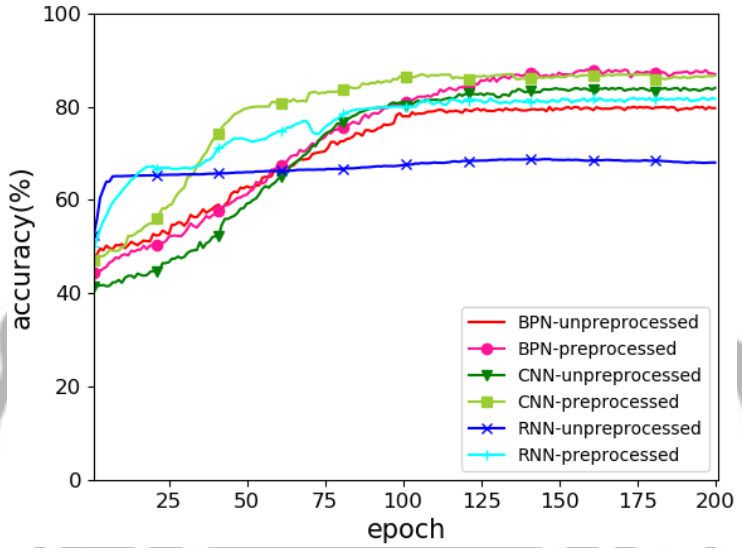


Figure 10. Accuracies of identifying pollution source when the pollution source is individually placed at node5 and node7. Data of PM10 are collected at node7.

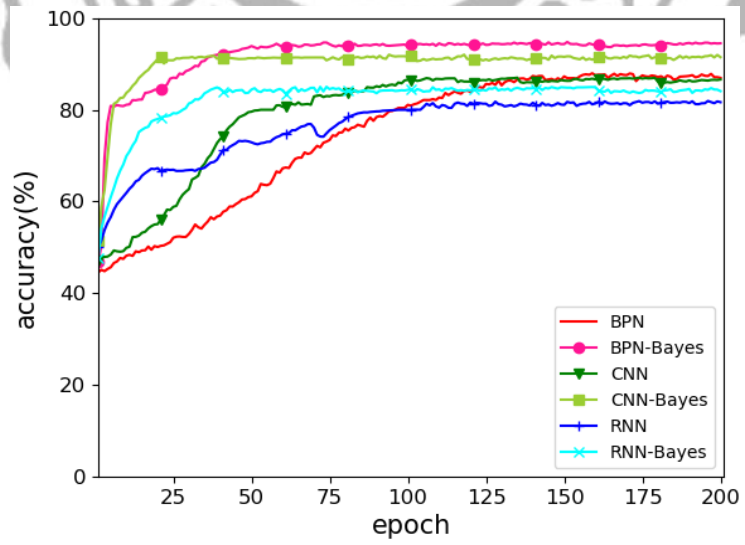


Figure 11. Accuracies of air-pollution-source identification before and after Bayesian Optimization.

Table 4. Comparison of model performance (before data processing; after data processing; Bayesian optimization).

Model	**BPN**			**CNN**			**RNN**		
Step	before	after	optimi	before	after	optimi	before	after	optimi
Accuracy (%)	79.8	87.6	94.2	83.7	86.5	91.3	68.3	81.7	84.1
Recall	0.57	0.61	0.88	0.64	0.65	0.85	0.36	0.55	0.78
Epoch	111	140	58	135	90	21	110	106	40

In fact, the concerned parameters can be adjusted to further improve the performance of the three neural network models. At the beginning of this stage, we manually adjusted parameters for these three models. It is time consuming since a neural network has a lot of parameters and each parameter's adjustable range is large, meaning that it is hard for us to manually tune these parameters one by one. In this study, some parameters, like strides and padding of CNN's convolutional-layer parameters and pooling-layer parameters which affects the number of extraction features, dropout of some neurons and initial values of parameters, e.g., forget bias and cell's state in LSTM Cell, are manually adjusted by us. Other hyperparameters are regulated by Bayesian optimization to optimize its objective function which in this study is employed as the loss function of the trained model, aiming to maximize identification accuracy. In other word, when training the model, those selected hyperparameters (see the first column of Table 5) are not necessary to be adjusted to the best on the convergence epoch (time). To identify whether a node is a pollution source, output layers adopt sigmoid to limit each of the five output values (corresponding to 5 IoT nodes placed at the center of the IoT field) to the range between 0 and 1. The purpose is to identify each of the five outputs of the five nodes as True or False. In fact, only the activation function of the hidden layer is tuned by Bayesian optimization.

The hyperparameters before and after Bayesian optimization are shown in Table 5. In order to achieve the best learning results, we manually adjusted the learning rate to a low value, i.e., 0.0001. After Bayesian optimization, the learning rate of the three models has increased a lot. Some of the other hyperparameters are significantly changed, e.g., number of BPN's nodes is increased from 100 to 296, and some remain unchanged.

Table 5. Hyperparameter selection before and after Bayesian Optimization.

Model	**BPN**		**CNN**		**RNN**	
Method	manual	Bayesian	manual	Bayesian	manual	Bayesian
Number of layers	10	6	2	3	1	2
Number of nodes	100	296	100	322	500	197
Learning rate	0.0001	0.0023	0.0001	0.0044	0.0001	0.0051
Activation function	sigmoid	tanh	sigmoid	Relu	sigmoid / tanh	sigmoid / tanh
Batch size	1	1	1	1	1	8
Optimizer	RMSProp	Adam	Adam	Adam	Adam	RMSProp

BPN

After Bayesian optimization, we found that the learning rate and the number of neurons of BPN (see the rows of "Learning rate" and "Number of nodes" in Table 5) are both increased, and the number of layers is reduced from 10 to 6. The activation function, optimizer and Batch size remain unchanged. The convergence of model training has been much earlier. The accuracy as shown in Table 4 is increased by 6.6% to 94.2% (from 87.6%), and the recall is increased from 0.61 to 0.88, both of which are individually the highest among the three nodes. But its training epoch (time) i.e., 58, is ranked the third.

CNN

The number of layers of CNN listed in Table 5, as the number of convolution layers, is 3 in this study. Number of nodes, as the number of neurons in the fully connected layer, is increased from 100 to 322 after Bayesian optimization. The learning rate is increased from 0.0001 to 0.0044, and the activation function is changed from sigmoid to Relu. Batch size (i.e., 1) and optimizer (i.e., Adam) are not changed. As illustrated in Table 4, the accuracy of CNN apparently is increased by 4.8% to 91.3% (from 86.5%), and the recall is increased from 0.65 to 0.85. Both are slightly lower than those of BPN. However, its convergence, i.e., 21, is the fastest among the three, probably because of the use of Relu (see Table 5). Generally, the convergence epoch (time) of Relu is faster than that of sigmoid. But because the number of layers that we use is only 2, the biggest possibility is the learning rate or applicability of CNN itself to the given data.

RNN

The number of layers of RNN refers to the number of layers of LSTM. We increase it to 2. The number of neurons, learning rate and Batch size are increased to 197, 0.0051 and 8, respectively. The activation functions of LSTM on the internal gates are still sigmoid and tanh. But the optimizer is changed to RMSProp. The data we collected may not be completely suitable for RNN, snice after optimization, the accuracy only increases 2.4% to 84.1% (from 81.7%). But its recall is increased to 0.78 from 0.58. On the other hand, probably because of increasing its learning rates and reducing number of neurons, its convergence (Epoch) has been significantly improved, i.e., from 106 to 40. However, its accuracy and recall are still the worst among the three models. Literature indicates that RNN has good analytical ability for sequential data [25] and good performance in air-pollution-concentration prediction [39]. This may be the reason why RNN is suitable for the condition when data has been collected for a long time, the collection region is wide, and wind speed and wind direction are stable. However, in this

experiment, it probably is that the data is collected in a small area, the data collection time is not long enough, and the wind direction and the smoke (pollution) flow are unstable. Consequently, RNN did not perform well, even after data preprocessing and hyperparameters optimization. When the region and data volume of this study are expanded, perhaps the performance of RNN will be better than what it has in this study.

In Table 4, we can see that the accuracies of these three neural network models have been improved after Bayesian optimization, their training's convergence epochs (times) are shortened, and recalls are significantly increased. It means that we can more effectively find the location of the pollution source, thus not only improving model's identification performance, but also saving a lot of time and energies for pollution-source identification. If originally one of the models does not perform well, the improvement is often obvious.

4.3 Analysis of Two Pollution Sources

In reality, air pollution often comes from several pollution sources simultaneously. This also shows that when identifying pollution sources, it is necessary to test whether an employed model can effectively detect multiple sources. But how to differentiate which pollution concentration is produced by which source is an engineering challenge. In fact, it is hard for us to obtain the data that clearly identifies multiple pollution sources and then train the model with these data. Therefore, in the third experiment, we wish to find the second pollution source given different experimental conditions. Three dual-source models (D-model for short) will be trained. The first was trained by using the data collected when there is only one pollution source. The second is that D-model was trained and tested by using the mixed data of single pollution source and two pollution sources. The third reuses the first model, but checks the probability that a node can be a pollution source.

4.3.1. Training a Model with Data of Single-Pollution Source

In fact, the first D-model is the one trained by using single-pollution source. But this time, we use it to identify two pollution sources. Table 6 shows that the accuracies of BPN, CNN and RNN are 58%, 65%, and 39%, respectively. The recalls of the three are all lower than 0.3. CNN occasionally finds two pollution sources. But in most cases, the accuracies are low. The reason is that the pollutant diffusion of two sources is different from that of single source. For example, in a single pollution source, the nodes that did not detect pollution may now discover polluted air in a D-model environment. Because the model is trained on a single-pollution source, the weights of the neurons used by the machine are only suitable for finding one pollution source. So it is a little hard to be directly applied to identify more pollution sources.

To conquer this problem, on each experiment, we check to see whether a node is a pollution source or not, rather than identifying two pollution sources. The activation function of the output layer is sigmoid. Consequently, the model can effectively analyze collected data only when the pollution diffusion of two pollution sources is similar to that of a single source. In fact, the machine can sometimes identify one of the two pollution sources, but in most cases, the model does not work well.

Table 6. The performance of BPN, CNN and RNN for identifying two pollution sources when they are trained with the data collected for single pollution source and tested with the data collected for two pollution sources.

Model	BPN	CNN	RNN
Accuracy (%)	58%	65%	39%
Recall	0.15	0.28	0.11

4.3.2. Training a Model with Data of Single Pollution Sources and Two Pollution Sources

As mentioned above, we individually train and test the three neural network models with single pollution source and two pollution sources. But in the test phase, we did not specify the number of pollution source that the model should identify. What we have done is only checking to see whether a node is a pollution source or not, through sigmoid, and then comparing the identification results with the actual locations of pollution sources.

Table 7 shows the accuracies (recalls) of BPN, CNN and RNN which are 88.2%, 87.6%, and 79.1% (0.81, 0.84 and 0.66), respectively. In particular, BPN and CNN have achieved certain performance in identifying pollution source. Although the accuracies of CNN are slightly lower than those of BPN, its recall, i.e., 0.84, is slightly higher than BPN's, i.e., 0.81, meaning that compared with BPN, the probability that CNN may identify a node which is not a pollution source as a pollution source, i.e., False positive, is higher than that of BPN. But CNN can also identify pollution sources more effectively than BPN and RNN since CNN's epoch (see Table 4) and recall (see Table 7) are better than BPN's.

Table 7. The performance of BPN, CNN and RNN for the identification of two pollution sources when they are trained and tested with the mixed data collected for single pollution source and two pollution sources.

Model	BPN	CNN	RNN
Accuracy (%)	88.2%	87.6%	79.1%
Recall	0.81	0.84	0.66

4.3.3. Identification the Most Likely Position of a Pollution Source

In the third D-model, we maintain the same data preprocessing and Bayesian optimization steps, and the activation function of its output layer is changed to softmax from their originated ones. We also change the algorithm for accuracy calculation. Instead of judging whether a node is pollution source or not, we check to see whether a node is the most likely to be a pollution source or not. If the model accurately identifies at least one pollution source when there are two pollution sources, we consider that this identification is success.

We first use the data collected when there is only one pollution source to train and testing a model, and then input two pollution sources for testing. The experiment process is the same as first D-model, but the activation function of output layer, i.e., sigmoid originally, is now substituted by softmax. The identification accuracy is defined as

$$\text{Identification accuracy with two pollution sources} = \frac{C}{D} \tag{5}$$

where *C* is the number of correct identification and *D* is the total number of identification. Note that Eq.3 is used to identify whether a node is a pollution source, and each tested data has 5 identification results, each of which is yes or no. Eq.5 is to determine whether a node is probably a pollution source or not based on tested data, no matter whether the data is a single source or two sources. Therefore, each tested record has only one identification result, i.e., yes or no. In other words, if the location of a node is the same as the location of a single pollution source or the location of one of the two pollution sources, we consider that the identification result is correct, and C is then increased by 1.

The test results are shown in Table 8. The identification accuracies with two pollution sources of BPN, CNN, and RNN are 79.1%, 82.6%, and 71.3%, respectively. Among them,

CNN has the highest accuracy. BPN's performance ranks the second. With this learning method, a model can effectively find the most likely pollution source given different numbers of pollution sources. Recall in the third D-model is meaningless since sometimes correct identification is only identifying a part of the two pollution sources. It is hard for us to define whether it is a successful recall or a fair recall.

Table 8. The performance of BPN, CNN and RNN when they are trained with the data collected for single pollution source and tested with the data collected for two pollution sources to identify the most likely position of a pollution source.

Model	BPN	CNN	RNN
Accuracy (%)	79.1%	82.6%	71.3%
Recall	-	-	-

V. Conclusion and Future Work

We originally envisioned the use of air diffusion models to help machines identify pollution sources. Therefore, in order to perform calculations quickly, we use the Gaussian diffusion model to simulate the diffusion of pollutants. But the results of the analysis are not as expected. Although the Gaussian diffusion model is fast, its accuracy is poor. We think that it is not suitable for air pollution source identification. When wanting to identify the location of the pollution source through the diffusion of pollutants, we need more accurate analysis results. Therefore, it is necessary to use a more effective model and more parameters to do this experiment.

Machine learning can really help us to find out air pollution source. Here, we confirm that preprocessing the collected data and employing Bayesian optimization can effectively increase the efficiency of machine learning and accuracies of pollution source identification. At the same

time, we also found that it is quite difficult to predict multiple pollution sources through limited data. The result is that they cannot effectively find two pollution sources when trained by only using the data of single pollution source.

So we use two other methods. One is to directly utilize the mixed data of single pollution source and two pollution sources for training and testing. The other is to use Softmax to find the most likely source of pollution in various types of data. They have performed well and we found that sometimes BPN has better accuracies than CNN and RNN individually have, but CNN has better accuracies and epochs (times) of convergence than BPN and RNN individually have in most cases. Among the two pollution sources, the efficiency of BPN is worse than that of CNN. In other words, CNN is relatively suitable for analyzing the source of air pollution in a grid field. We think that CNN can identify the relationship between position of nodes and wind direction from a small amount of data. This is why CNN performs the best.

We confirm the feasibility of applying the Gaussian diffusion model to identify air pollution sources and propose a model prototype for identifying pollution sources by utilizing CNN. In the future, we wish to gradually expand the region of the IoT field and increase the number of features, such as temperature, atmospheric pressure and other pollutant concentrations. We would also like to integrate our models with air diffusion models to analyze the diffusion of air pollution and calculate the concentration of pollutant for the place in which no IoT nodes are placed. Finally, the pollution data will be collected from a large-scale area, such as small air monitoring stations, wishing that this model can identify air pollution sources of our living environment in a real-time manner so as to effectively monitor and improve air pollution for people. These constitute our future studies.

Reference

[1] Q. Sun, X. Hong and L.E. Wold, “Cardiovascular effects of ambient particulate air pollution exposure,” *Circulation*, vol. 121, pp. 2755- 2765, June 2010.

[2] World Health Organisation, “Public health, environmental and social determinants of health,” https://www.who.int/phe/en/.

[3] U. Gehring, A.H. Wijga, M. Brauer, P. Fischer, J.C. de.Jongste, M. Kerkhof, M. Oldenwening, H.A. Smit, B. Brunekreef, “Traffic-related air pollution and the development of asthma and allergies during the first 8 years of life,*American Journal of Respiratory and Critical Care Medicine*, vol. 181, no. 6, pp. 596–603, December 2009.

[4] L.E. Plummer, S. Smiley-Jewell and K.E. Pinkerton, “Impact of air pollution on lung inflammation and the role of toll-like receptors,” *International Journal of Interferon, Cytokine and Mediator Research*, vol. 4, pp. 43–57, June 2012.

[5] K. Straif, A. Cohen and J. Samet, “Outdoor air pollution a leading environmental cause of cancer deaths,” *IARC Scientific Publication No. 161: Air Pollution and Cancer*, October 2013.

[6] P. A. Solomon, P. Gehr, D.H. Bennett, R.F. Phalen, L.B. Méndez, B. R.Rutishauser, M.Clift, C. Brandenberger and C. Mühlfeld, “Macroscopic to microscopic scales of particle dosimetry: From source to fate in the body,” *Air Quality, Atmosphere & Health*, vol. 5, no. 2, pp. 169–187, June 2012.

[7] D. Hu and J. Jiang, “PM2.5 pollution and risk for lung cancer: A rising issue in China,” *Journal of Environmental Protection,* vol. 5, no. 8, pp. 731–738, June 2014.

[8] J. Jin, J. Gubbi, S. Marusic and M. Palaniswami, “An information framework for creating a smart city through Internet of Things,” *IEEE Internet Things Journal*, vol. 1, no. 2, pp. 112–121, April 2014.

[9] PM2.5 Open Data Portal. Accessed: Jan. 21, 2018. [Online]. Available: http://pm25.lass-net.org/en/

[10] B. Rajesh, A. Agarwal and K.A. Saravanan, ‘‘Proficient modus operandi for scrutinize air pollution using wireless sensor network,’’ *International Conference on Circuits, Power and Computing Technologies*, pp.1312-1316, March 2014.

[11] X. Chen, X. Liu and P. Xu, ‘‘IOT- based air pollution monitoring and forecasting system’’ *International Conference on Computer and Computational Sciences*, pp. 257-260, December 2015.

[12] G.A. Briggs, “Diffusion estimation for Small emissions,” *National Oceanic and Atmospheric Administration*, *Oak Ridge*, *Tenn. (USA). Atmospheric Turbulence and Diffusion Lab*, May 1973.

[13] S.R. Hanna, G.A Briggs and R.P. Hosker, “Handbook on Atmospheric Diffusion,” *National Oceanic and Atmospheric Administration*, *Oak Ridge*, *TN (USA). Atmospheric Turbulence and Diffusion Lab*, January 1982.

[14] T.K. Flesch and J.D. Wilson, “Backward-time Lagrangian stochastic dispersion models and their application to estimate gaseous emissions,” *Journal of Applied Meteorology*, pp.1320-1332, June 1995.

[15] T.K. Flesch et al., “Deducing ground-to-air emissions from observed trace gas concentrations: a field trial,” *Journal of Applied Meteorology*, pp.475-484, March 2004.

[16] A. Cortis, C.M. Oldenburg, “Short-range atmospheric dispersion of carbon dioxide,” *Boundary-Layer Meteorology*, pp.17-34, October 2009.

[17] M. Pontiggia, M. Derudi, V. Busini and R. Rota, “Hazardous gas dispersion: a CFD model accounting for atmospheric stability classes,” *Journal of Hazardous Materials*, vol. 171, issue 1-3, pp.739-747, November 2009.

[18] D.W. Byun, J. Young, G. Gipson, J. Godowitch, F. Binkowsky, S. Roselle, B. Benjey, J. Pleim, J.K.S. Ching, J. Novak, C. Coats, T. Odman, A. Hanna, K. Alapaty, R. Mathur, J. McHenry, U. Shankar, S. Fine, A. Xiu and C. Lang,“Description of the Models-3 Community Multiscale Air Quality modeling systemin,” *78th Annual Meeting of the American*

Meteorological Society, pp.264–268, January 1998.

[19] J.C. Linford, J. Michalakes, M. Vachharajani and A. Sandu, "Automatic generation of multicore chemical kernels," *IEEE Transactions on Parallel and Distributed Systems*, pp.119 - 131, May 2010.

[20] D. Ma and Z. X. Zhang, "Contaminant dispersion prediction and source estimation with integrated Gaussian-Machine learning network model for point source emission in atmosphere," *Journal of Hazardous Materials*, vol. 311, pp.237-245, July 2016.

[21] M. Abadi et al., "TensorFlow: A system for large-scale machine learning TensorFlow: A system for large-scale machine learning," *12th USENIX Symposium on Operating Systems Design and Implementation*, pp. 265-284, May 2016.

[22] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol.2, issue 5, pp. 359-366, March 1989.

[23] Y. Kim, "Convolutional neural Networks for sentence classification," *the Conference on Empirical Methods in Natural Language Processing. Stroudsburg*, pp. 1746-1751, October 2014.

[24] T. Lin , B.G. Horne, P. Tino and C.L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Transactions on Neural Networks*, vol.7, no.6, pp.1329–1338, November 1996.

[25] T. Lin, B.G. Horne, P. Tino and C.L. Giles, "Learning to forget: continual prediction with LSTM," *Neural Computation*, vol.7, no.6, pp.2451-2471, October 2000.

[26] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro and D. Sculley, "Google Vizier: A Service for Black-Box Optimization," *the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1487-1495, August 2017.

[27] Tobe, Open-source implementation of Google Vizier for hyper parameters tuning, https://github.com/tobegit3hub/advisor, 2018.

[28] Y. Chen, A. Huang and Z. Wang et al., "Bayesian Optimization in AlphaGo," *Computing*

Research Repository, vol. abs/1812.0, December 2018.

[29] F. Dernoncourt and J.Y. Lee, "Optimizing neural network hyperparameters with Gaussian processes for dialog act classification, "*IEEE Spoken Language Technology Workshop*, pp.406-413, September 2016.

[30] M. Markiewicz, ''A review of mathematical models for the atmospheric dispersion of heavy gases. Part I. A classification of models, ''*Ecological Chemistry and Engineering Society*, vol. 19, no. 3, pp. 297–314, July. 2012.

[31] S. Ma and X. Jin, ''Simulation on the emission of SO2 and NOx from coal-fired power plants using MM5-SMOKE-CMAQ,'' *Asia-Pacific Power and Energy Engineering Conference*, pp. 190-195, April 2010.

[32] S. Ma, ''Seasonal simulation of ozone by air quality model-CMAQ in the great lakes," *4th International Conference on Bioinformatics and Biomedical Engineering*, https://ieeexplore.ieee.org/document/5516066, July 2010.

[33] D. Ma, J. Deng and Z. X. Zhang, ''Comparison and improvements of optimization methods for gas emission source identification," *Atmospheric Environment*, vol. 81, pp.188-198, December 2013.

[34] R. Keresztes and E. Rapo, ''Statistical analysis of air pollution with specific regard to factor analysis in the Ciuc basin, Romania," *Studia Universitatis Babes-Bolyai, Chemia*, vol. 62, issue 3, pp.283-292, September 2017.

[35] K. B. Shaban, A. Kadri and E. Rezk, ''Urban air pollution monitoring system with forecasting models," *IEEE Sensors Journal,* vol.16, pp. 2598-2606, April, 2016.

[36] L. Contreras and C. Ferri, ''Wind-sensitive interpolation of urban air pollution forecasts," *Procedia Computer Science*, vol.80, pp. 313 – 323, December 2016.

[37] L. Contreras-Ochando and C. Ferri, ''airVLC: An application for visualizing wind-sensitive interpolation of urban air pollution forecasts," *IEEE International Conference on Data Mining Workshops*, pp.1296 – 1299, December 2016.

[38] A. Kurt and A.B Oktay, ‘‘Forecasting air pollutant indicator levels with geographic models 3 days in advance using neural networks,” *Expert Systems with Applications*, vol. 37, pp. 7986-7992, December 2010.

[39] A. Chakma, B. Vizena, T. Cao, J. Lin and J. Zhang, ‘‘Image-based air quality analysis using deep convolutional neural network,’’ *IEEE International Conference on Image Processing*, pp. 3949-3952, September 2017.

[40] Y.T. Tsai, Y.R. Zeng and Y.S. Chang, “Air pollution forecasting using RNN with LSTM,” *IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, pp.1074-1079, October 2018.

[41] R.A. Bahari, R.A. Abbaspour and P. Pahlavani, ‘‘Prediction of PM2.5 concentrations using temperature inversion effects based on an artificial neural network,’’ *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2. pp. 73–77, 2014.

[42] D.J. Lary, F.S. Faruque, N. Malakar, A. Moore, B. Roscoe, Z.L. Adams, Y Eggelston, ‘‘Estimating the global abundance of ground level presence of particulate matter (PM2.5),’’ *Geospatial Health*, vol. 8, no. 3, pp. 611–630, December 2014.

[43] X. Feng, Q. Li, Y. Zhu, J. Hou, L. Jin and J. Wang, “Artificial neural networks forecasting of PM2.5 pollution using air mass trajectory based geographic model and wavelet transformation,” *Atmospheric Environment*, vol. 107, pp. 118-128, April 2015.

[44] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang and T. Li, ‘‘Forecasting fine-grained air quality based on big data,’’ 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2267–2276, August 2015.

[45] B.T. Ong, K. Sugiura and K. Zettsu, ‘‘Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5,’’ *Neural Computing and Applications*, vol. 27, no. 6, pp. 1553–1566, August 2016.

[46] H.D. Shannon, G.S. Young, M.A. Yates, M.R. Fuller and W.S. Seegar, “Measurements of thermal updraft intensity over complex terrain using american white pelicans and a simple boundary-layer forecast Model,” *Boundary-Layer Meteorology*, vol.104, issue 2, pp.167-199, August 2002.

[47] M.L. Barad, “Project prairie grass, a field program in diffusion,” *Air Force Cambridge Centre, Massachusetts USA*, July 1958.

[48] C. Alippi and M. Roveri et al., “Virtual k-fold cross validation: An effective method for accuracy assessment,” *The 2010 International Joint Conference on Neural Networks,* pp.1-6, October 2010.