A Novel Predictor for Exploring PM2.5 Spatiotemporal Propagation by Using Convolutional Recursive Neural Networks

Hsing-Chung Chen[1,2], Karisma Trinanda Putra[1,3], Chien-Erh Weng[4*], Jerry Chun-Wei Lin[5]

[1]Department of Computer Science and Information Engineering, Asia University, Taiwan
[2]Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan
[3]Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Indonesia
[4]Department of Telecommunication Engineering, National Kaohsiung University of Science and Technology, Taiwan
[5]Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Norway
cdma2000@asia.edu.tw, karisma@ft.umy.ac.id, ceweng@nkust.edu.tw, jerrylin@ieee.org

Abstract

The spread of PM2.5 pollutants that endanger health is difficult to predict because it involves many atmospheric variables. These micro particles could spread rapidly from their source to residential areas, increasing the risk of respiratory disease if exposed for long periods. However, the existing prediction systems do not take into account the geographical correlation among neighboring nodes spatially and temporally resulting in loss of important information, lack of PM2.5 propagation resolution, and lower forecasting accuracy. In this paper, a novel scheme is proposed to generate propagation heat maps of PM2.5 prediction by using spatiotemporal datasets. In this scheme, the deep learning model is implemented to extract spatiotemporal features on these datasets. This research was conducted by using the dataset of air quality monitoring systems in Taiwan. Moreover, the robust model based on the convolutional recursive neural network is presented to generate the propagation maps of PM2.5 concentration. This study develops an intelligence-based predictor by using Convolutional Recursive Neural Network (CRNN) model for predicting the PM2.5 propagation with uncertain spread and density. It is also one of key technologies of software and hardware co-design for massive Internet of Things (IoT) applications. Finally, the proposed model the proposed model provides accurate predictive results over time by taking into account the spatiotemporal relationship among sensory nodes in order to give a prediction solution for the massive IoT deployment based on green communication.

Keywords: PM2.5 propagation, Spatiotemporal, Convolutional recursive neural network, AI

1 Introduction

Particulate matter (PM) below 2.5 um, which is called PM2.5, is one of the biggest urban problems and has become the most dangerous source of air pollutants. PM2.5 is very dangerous for health especially for vulnerable groups, *e.g.,* infants, children, pregnant women, and the elderly. A high concentration of PM2.5 is one of the factors that cause dangerous disease *e.g.,* heart disease, respiratory infections, cancer, and chronic lung disease [1]. PM2.5 is a micro particle that could aggravate the respiratory disease more quickly because it can settle to the respiratory tract of the bronchi and alveoli [2]. PM2.5 is more dangerous than particulate matters below 10 (PM10) because it is attached to the alveoli instead of filtered in the upper respiratory system. These particles reduce the ability of the lungs to absorb oxygen from the air. Due to its ultra-light odorless physical characteristics, the long-term impacts will be unrecognized by the residents [3]. PM2.5 concentration is difficult to detect accurately, has a high number of outliers, and is highly dependent on other atmospheric parameters [4]. Moreover, it is hard to predict the ever-changing PM2.5 concentrations because the features are highly influenced by other variables *e.g.,* geographical location, wind direction, temperature, humidity, and other sources of pollutants [5].

WSN-based environmental sensing has been widely used to measure the PM2.5 concentrations accurately using a large number of sensory nodes. However, measuring air pollution on a scattered massive scale sensory node becomes a major challenge. A large number of end sensory nodes increases the occurrence of sparse data. Compressive sensing algorithms [6-7] could be used to improve the performance of the prediction system while reducing sparse data generation. Although the spatial propagation of air pollutants could be monitored in a higher resolution, there is a possibility that some measurement errors are obtained from millions of data, resulting in inconsistent predictions. On the other hand, the data has a sequential feature that changes over time. These features could be extracted and predicted by an Artificial Intelligence (AI) model. The prediction model could use the spatiotemporal data pattern tied together in terms of space and time domain [8-9] in order to improve theoretically the accuracy. By using deep learning model, this research proposes a novel approach to explore spatiotemporal patterns inside the dataset.

In Taiwan, the government has implemented PM2.5 measuring sensors in every weather station distributed in each district. Cooperation between the government and WSN manufacturers has also generated many sensor nodes installed with a large but uneven distribution. This collaboration generates a pollutant measurement system *i.e.,* 'Airbox' devices [10] that have been implemented and its record with a certain period can be accessed in real-time via Internet. The

*Corresponding Author: Chien-Erh Weng; E-mail: ceweng@nkust.edu.tw
DOI: 10.53106/160792642022012301017

data is mined from the 'Airbox' to provide earlier and accurate predictions.

The spatiotemporal pattern could be used to predict the distribution of pollutants in the future [11]. The direction of pollutant distribution could be predicted accurately using the spatiotemporal dataset. In general, the proposed framework uses a Recursive Neural Network (RNN) to extract short and long-term patterns from the PM2.5 dataset. Moreover, the spatial patterns on the dataset are learned by using Convolutional Neural Network (CNN). The combination of CNN and RNN has been shown to be used to record many and complex pre-trained patterns [12]. Therefore, this study presents a novel Convolutional Recursive Neural Network (CRNN) model to predict the propagation of PM2.5 from time to time.

The contributions of this paper are summarized below.

1) The novel predictor model capable of servicing large-scale sensor networks with thousands of sensor nodes is developed to generate high-resolution PM2.5 propagation maps.
2) This framework utilizes CNN to extract spatial features of a WSN and RNN to learn the temporal features from data sequences so that the correlation relationship between measurement results could be maintained. Moreover, the spatiotemporal correlation relationship is extracted as a feature in order to provide better forecasting results than the previous approach.
3) This framework supports reconfigurable network deployment, *e.g.,* reduction, addition, or topology modification to increase measurement precision without the need for repetitive training processes.
4) The quantitative analyses in terms of prediction accuracy and error rate are performed by using the real-time 'Airbox' dataset obtained from 268 sensor nodes located in Central Taiwan. Moreover, a qualitative analysis of the results is also presented.
5) The geospatial heat maps that describe predictions of the air pollutants propagation are provided for the next day, where the prediction model only uses PM2.5 data variables without involving other atmospheric variables.

The remainder of this paper is described as follows. Section II reviews the previous works in terms of PM2.5 prediction. Section III provides basic methods to build the prediction model. Section IV describes in detail the proposed model. Section V shows the experiments and the results. Furthermore, a discussion is provided in Section VI. Finally, a conclusion is summarized in Section VII.

2 Related Works

PM2.5 is a micro particle that floats easily in Earth's atmosphere. The propagation of PM2.5 is closely related to changes in atmospheric variables, *e.g.,* wind speed and wind direction. Wind speed and direction data have a high degree of randomness and always change over different periods [13]. It affects the direction, area, and speed of this pollutant propagation [14]. Until now, there are only a few models that describe the propagation of micro particles in a predictor model that includes time and space domains [15-17]. Furthermore, the existing models have not been implemented to massive scale sensor networks. The probability of the error rate increases, with the increasing number of sensor nodes installed on a system [18]. This type of pollutant data always changes depending on human activities, atmospheric variables, and geographical position. To predict PM2.5 concentration, a previous study uses clustered data to simplify the process because it involves massive data from thousands of sensor nodes. By using clustering methods, the data is processed efficiently. It is very useful for large-scale predictor modeling, at the expense of the resolution of the prediction results. On the other hand, a linear regression analysis can be used to design a forecasting model. However, this method cannot capture too many and complex features [19].

To model a complex signal with a high degree of randomness, a previous study [20-21] utilized Neural Networks (NN). However, the experiment only used a small one-dimensional dataset without spatiotemporal feature extraction. Moreover, other researchers used Extreme Machine Learning (ELM) to predict air pollution [22]. However, this approach could only be used for a small dataset with features which are not too complex. The more complex the learned features, there are higher the chances that the NN cannot achieve convergent learning. These complex features usually are found evenly in most areas of the signal. The discovery of the RNN model raises new possibilities related to forecasting sequential data even those with high complexity features. Furthermore, entering the era of deep learning, many predictive models are starting to emerge and it produces better accuracy than conventional NN.

Multiple variants of RNN *e.g.,* Gated Recurrent Unit (GRU) network and Long Short-Term Memory (LSTM) network are proven to be used to build sequential data prediction systems [23-24]. The temporal pattern shows very strong data relations between the present data and the previous data. This correlation is learned by GRU and LSTM networks and becomes knowledge *i.e.,* manifested in the convergent weights on each layer. In addition, the data recorded from a WSN has special features spatially and temporally. The temporal features could be extracted using RNN, while the spatial features could be recorded using CNN. Previous research discussed the potential of CNN-LSTM to predict PM2.5 [25]. However, this prediction is only based on one-dimensional data so that the spatial relationship of the sensor networks cannot be extracted extensively. The proposed model is capable of recording a 2-dimensional dataset with a feature extraction that can record spatiotemporal patterns. Furthermore, in order to extract high complexity features, a large number of neuron networks are required which are arranged in stratified layers. The proposed model utilizes a deep learning scheme that combines CNN and RNN in several cascade layers to extract spatiotemporal dataset. The spatiotemporal model provides accurate predictions as evidenced in [26].

3 Methods

In this section, the general knowledge regarding to both convolutional and recursive NN is presented. At the beginning of this chapter, several types of recursive networks were introduced (*i.e.,* RNN, GRU, and LSTM) followed by a convolutional network (*i.e.,* CNN). In the end, a convolution CRNN model is presented to extract both spatial and temporal features simultaneously.

3.1 Recursive/Recurrent Neural Network (RNN)

RNN is a form of ANN architecture that is specifically designed to learn sequential data. It is usually used to process tasks related to time series data. RNN is quite widely used to solve sequential problems [27], *e.g.,* Natural Language Processing (NLP), speech recognition, machine translation, video classification, stock prediction, and weather forecasting. The main idea of the RNN architecture is to exploit sequential data structures by feeding back the ANN's output as recursive inputs. It means that the same operation is performed for each sequence element, with the output depending on the current input and the previous operation. In essence, RNN focuses on the nature of the data where the previous $x_{(t-1,t-2,\ldots,t-T)}$ and present time x_t affects the next time variable x_{t+1}. RNN does not discard information from the past. RNN extracts information from the past by looping through its architecture, which automatically stores information from the past into its recursive weights.

3.2 Gated Recurrent Unit (GRU)

The idea of designing a GRU network is that each iterative unit captures dependencies in different time scales adaptively [28]. It can be analogized to a rainfall prediction system in an area with two seasons, *i.e.,* dry season and rainy season. Information from the past about rainfall in the dry season will not contribute significantly to decision-making when the current conditions are in the rainy season. Inside GRU, the information flow control component is called a gate and the GRU has 2 gates, namely a reset gate and an update gate. The reset gate determines how to combine the new input with past information. Meanwhile, the update gates determine how much past information should be stored while reading/generating a sequence.

3.3 Long Short-Term Memory (LSTM)

LSTM is a type of RNN with the addition of a memory cell that could store information for a long time. LSTM is proposed as a solution to overcome the vanishing gradient in RNN when processing long sequential data. LSTMs are able to learn long-term dependencies that were previously a weakness in RNN. LSTMs also have repeating connections or chain-like structures. The difference between LSTM and RNN lies in the layers contained in each LSTM cell. In each LSTM cell, there are 3 Sigmoid functions and 1 Hyperbolic Tan function. For long-term dependency problems, LSTM could handle noise, distributed representation, and continuous values [29-30].

3.4 Convolutional Neural Network (CNN)

CNN is a type of NN with its inputs in the form of two-dimensional data so that the linear operations and weight parameters on CNN are different from NN. Inside CNN, linear operations use convolutional operations, while weight is no longer one-dimensional, but in four dimensions which is a collection of convolutional kernels. CNN is inspired by the Visual Cortex, which is the part of the brain that processes information in visual form. By using a multi-layer (deep) architecture, CNNs can be trained to extract complex features in a dataset [31]. CNN can capture spatial and temporal dependencies in an image when operated with relevant filters so that it can be used to predict a short sequence of 2-dimensional data [32]. To provide a better understanding of a complex pattern, CNN is usually designed with many interconnected layers, so it is commonly called a deep learning architecture.

3.5 Convolutional Recursive Neural Network (CRNN)

The traditional convolutional layer extracts feature from the data by applying non-linearity to the activation function of the input. CRNN upgrades this feature extraction process especially for the case of sequential data, by inputting the data into the CNN then using the output as inputs of RNN layers [33]. This architecture exploits the fact that a window containing multiple frames of sequential data is a temporal feature that might encapsulate valuable information. Meanwhile, spatial features are recorded in the convolution layer which is directly connected to the input layer. In addition, CRNN model can be used to predict the dataset with long sequence format, *e.g.,* internet sentiment analysis [34] and public opinion analysis [35].

4 Design of Prediction Model Based on CRNN

The basic concepts of designing a predictor model on a massive scale sensor network are described in this section. The initial section describes the process of data collection and preprocessing. Then, the characteristics of an RNN model are explored. Finally, the proposed CRNN model is described to extract spatiotemporal features in the dataset.

4.1 Tools and Dataset

The dataset is obtained from sensor nodes as part of the 'Airbox' sensor network that provides real-time PM2.5 monitoring services. The system has been implemented throughout Taiwan and includes thousands of community-installed sensor nodes and hundreds of government-owned nodes. This air quality monitoring system is hereinafter referred to as the 'Airbox'. The proposed scheme is shown in Figure 1. This system does not completely change the already implemented 'Airbox' system. This schema updates the server node by implementing AI technology for a more precise forecasting model. The hardware specifications are shown in Table 1. As a limitation, only data from several of 'Airbox' sensory nodes (*i.e.,* 268 sensor nodes scattered across Central Taiwan) are collected. Central Taiwan is suitable as a research case because as referred to [1], this area reflects the condition of Taiwan as a whole. It is also home to the third-largest coal-fired power plant in the world, which produces large emissions of carbon and micro particles.

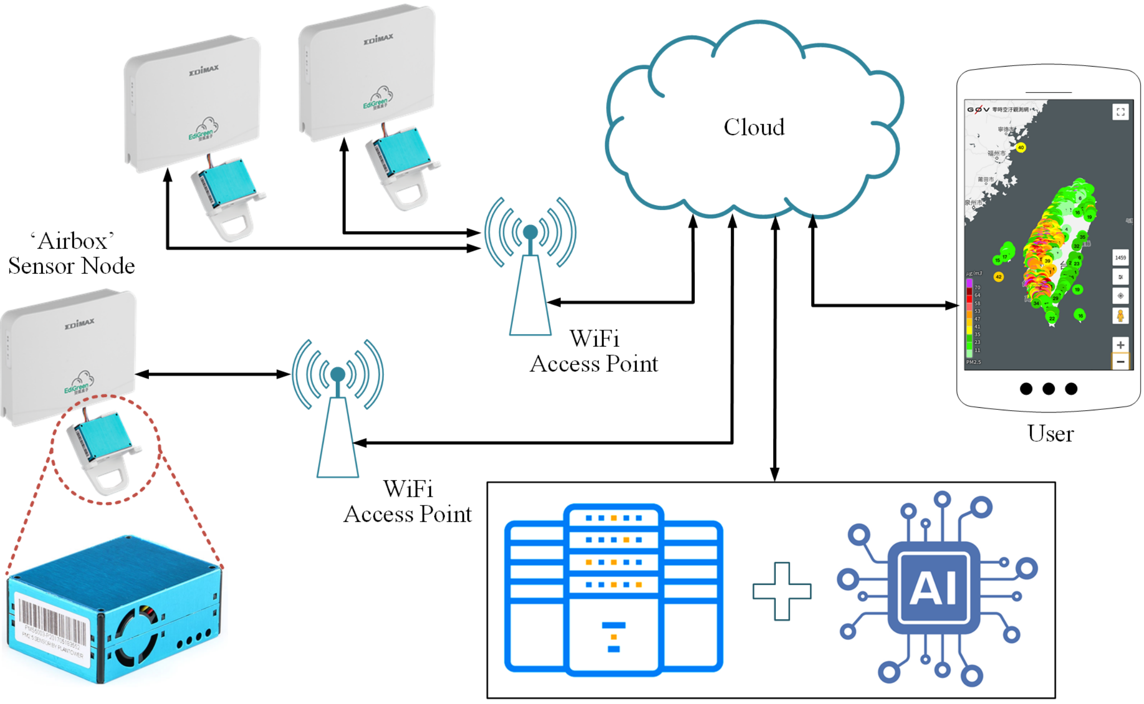

Figure 1. The proposed schema utilizes WSN that served by PM2.5 sensor nodes named 'Airbox' combined with AI cloud computing

Table 1. The specification of sensor nodes and server node used in this research

PM2.5 Sensor Node		Server Node	
Size	: 148.4mm x 111.5mm x 45mm	Processor	: Dual 20-Core Intel Xeon E5-2698 2.2 GHz
Connection	: WiFi IEEE 802.11b/g/n	RAM	: 256 GB
Temperature sensing range	: 0~60°C, Accuracy: ± 1 °C	GPU	: NVIDIA Tesla P100 (3584 CUDA Cores)
Humidity sensing range	: 0~100%RH, Accuracy: ± 5%	GPU memory	: 32GB HBM2
Measurement range	: minimum 0.3μm	OS	: Ubuntu 64-bit
Measurement efficiency	: 50% @ 0.3um, 98% @ >= 0.5 um	Environment	: Python with Keras Tensorflow backend
Power supply	: Micro USB port x 1 DC 5V	Library	: numpy, pandas, pyplot, and folium

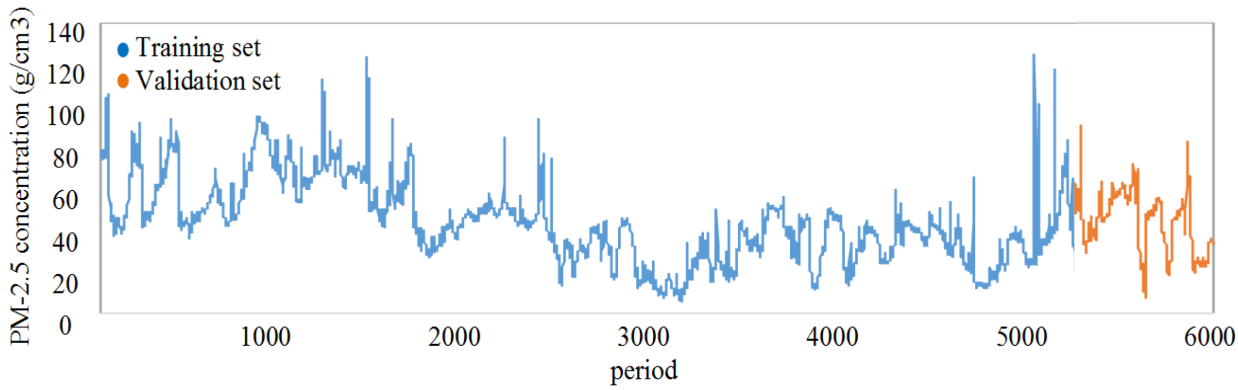

Figure 2. A single dataset from a sensor node measured in one month with a 5-minutes sampling time. 90% of it is used as training data and the remaining 10% rest as testing data

PM2.5 dataset was collected for one month *i.e.,* recorded in September 2020 with a sampling rate of every 5 minutes. Experiments were performed on a dataset (8.7MB) of 268 sensor nodes. Each sensor generates approximately 32KB of data per month of measurement. However, only PM2.5 data was used to evaluate the proposed framework. The dataset from a sensor node is visualized in Figure 2. Visually, it can be seen that the data has a rather random pattern but still has a repetition pattern every day. Therefore, to simplify the experiments, the dataset is compressed into 2 hours sampling rate by averaging the data each 2 hour. For the experiment, 90% of the dataset is used as training data and 10% rest as testing data. The evaluation process is carried out using training losses, accuracy, and RMSE.

4.2 Design of A Simple Predictor Using RNN layers

The design of the RNN model is carried out using the python programming language with the support of the TensorFlow library. In this section, each type of recursive layer is evaluated to decide the best layer and the optimal configurations that will achieve the best performance. A dataset from a single sensor was used as experimental material. The lowest losses and the best accuracy generated by the three types of recursive layers (*i.e.,* RNN, GRU, and LSTM) were presented. These approaches do not involve the extraction of spatiotemporal features. These approaches were compared to explore their characteristic (*i.e.,* losses, accuracy, and efficiency).

At first, the target of this forecasting system is determined, which is to forecast PM2.5 concentration with a sampling time of 2 hours for the next 24 hours with the past 1-day dataset. Each day will generate 12 data in a sequence. As shown in Figure 3, model compilation and model specification were defined. In accordance with the target, the number of input neurons is determined to be 12 x 2 days = 24 neurons. A compilation process is carried out to arrange these layers into a deep learning model. Furthermore, the process of arranging the layers into a model is carried out sequentially with each type of recursive layer having a different number of layers varying from 1 to 5. Between each layer, a dropout layer is inserted to prevent overfitting and also accelerate the learning process.

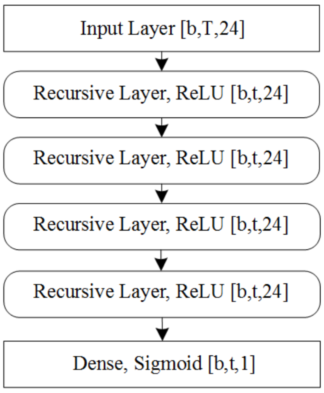

Figure 3. Design of a simple predictor using several recursive layers *i.e.,* RNN, GRU, and LSTM

Table 2. The different layer depth variations on a simple RNN affect the execution time without significantly increasing accuracy

Number of Layers	Number of Parameters	Training Time (s)	Losses	Accuracy
1	6,49	11	0.0143	0.7583
2	1,825	19	0.0084	0.8943
3	3,001	35	0.0011	0.9514
4	4,177	42	0.0025	0.9011
5	5,353	67	0.0012	0.9102

Table 3. The difference between a simple RNN, GRU, and LSTM by using different number of input channels

Recursive Layer	Number of Parameters	Input Channels	Losses	Accuracy
Simple RNN	4,177	1	0.0020	0.9156
	4,201	2	0.0018	0.9211
	4,225	3	0.0019	0.9180
	4,249	4	0.0025	0.9011
GRU	12,769	1	0.0015	0.9455
	12,841	2	0.0014	0.9493
	12,913	3	0.0018	0.9330
	12,985	4	0.0020	0.9190
LSTM	16,633	1	0.0011	0.9514
	16,729	2	0.0007	0.9673
	16,825	3	0.0015	0.9404
	16,921	4	0.0019	0.9211

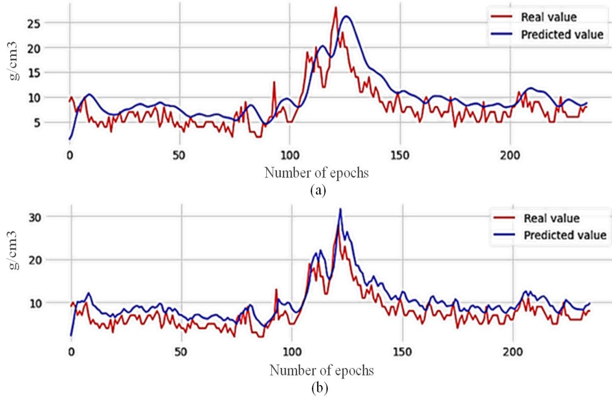

Figure 4. The results performed by using a recursive model *i.e.,* (a) GRU and (b) LSTM

From Table 2, it can be concluded that the 4-layer configuration provides optimal performance without sacrificing a lot of neurons. The use of more than 4 layers does not provide significant performance, thus decreasing computational efficiency. Then, this 4-layer configuration is used to design the proposed model. To see the best performance on the forecasting system, another experiment takes into account the measurement results of the closest nodes around the test node. In theory, the more data that is included in the calculation, it affects the accuracy of the system. Therefore, each input has a channel variation from 1, 2, 3, and 4 which represents the number of closest nodes that are used as modeling references. The output is 1 neuron which represents the next 2-hours prediction. Furthermore, because the network used is a recursive type, the input will be shifted and the output will represent the predicted results of the next hour. After compilation, the number of parameters in the LSTM and GRU network is 4 and 3 times more than that of on the simple RNN as shown in Table 3. The greater the number of parameters, the longer the training process will take. Likewise, the greater the number of layers, the greater the number of parameters. With the increasing number of parameters, the more local and global features that could be stored on it. It can be seen that the best accuracy is performed by LSTM. Even if nodes around the location is computed into the learning model, the performance does not improve significantly. Increasing the number of input channels actually increases the number of neurons without a significant contribution to accuracy. The prediction results look good visually and are able to replicate the real-world value shown in Figure 4. However, these results are obtained for forecasting one step ahead. To predict the next few steps by involving the results as inputs, this one-dimensional recursive model has not provided good performance.

4.3 Design of a Novel Predictor Framework Using CRNN Layers

The proposed design consists of 2 main parts *i.e.,* preprocessing and CRNN modules which are composed of CNN encoder, RNN, and CNN decoder, shown in Figure 5.

4.3.1 Preprocessing

Preprocessing converts raw data with an unequal distribution into structured data, which has strong tied in spatially and temporally. This preprocessing involves the geographical position of each node and its measurement time. That is, each data is collected and arranged based on both the time in sequence as well as the spatial information in geographic location according to Voronoi diagram shown in Figure 6, where Voronoi diagram [36] is a mathematical method to partition a plane into regions closing to each of a given set of objects. The data is collected by the node named 'Airbox' in this research, which the PM2.5 data is detected and generated by using five minutes sampling rate. The data is resampled every 2 hours for the last 2 days (24 sequences) as a training set, while 12 hours of subsequent data were used as ground truth. The dataset is processed by the model, then the results are the next 24 hours prediction values.

The dataset is not evenly distributed spatially. It can be difficult to feed unevenly distributed data directly to a deep learning model. Therefore, the data is divided spatially into several sectors H. Taiwan has a main island that stretches from North to South. For the experiment, Taiwan region is divided into small sectors with a resolution of 40 rows m and 40 columns n starting from coordinates (23.90, 120.37) to coordinates (24.45, 121.020). This configuration is chosen so that one sector only covers 1 to 10 nodes. If the resolution is higher, the number of neurons used will be bigger which will affect the computer's ability to complete the training process.

Each sector provides similar features for several sensor nodes. In one sector, it consists of several cells that represent the coverage area of each node as shown in Figure 5. Each sector is connected to 1 input neuron. Therefore, Equation (1) is needed to convert the values of several sensor node measurements into an aggregation value that represents the value of the sector. Moreover, the Voronoi diagram is used to calculate the contribution of a node in a sector. A Voronoi diagram is a division of the area of a plane into sections based on the distance from points on a specific subset of the area of the plane. The diagram divides a sector based on the position of the nodes in the plane into several Voronoi cells with a certain area. Each sector has a certain value and it forms a heat map value which represents the PM2.5 pollution level. Given sensor nodes in geographical position (i, j) inside certain sector $\zeta_{m,n}$. Equation (1) is used to compute the heat maps as inputs of the deep learning model.

$$\theta_{m,n} = \begin{cases} \sum x_{i,j} \dfrac{v_{i,j}}{V_{m,n}}; \; x_{i,j} \in \zeta_{m,n} \\ 0 \qquad ; \; x_{i,j} \notin \zeta_{m,n} \end{cases}, \tag{1}$$

where θ, x, v, V are heat map value, PM2.5 value, volume of Voronoi cell, and volume of sector respectively. In addition, heat map is a useful tool to see the activity of the research object with color indicators. This color represents how active, much, and intensely PM2.5 propagation behaves in environments with geographic position labelling.

4.3.2 Going Deeper into CRNN Model

A bottom-up problem implementation strategy is used to design the prediction model. The research that has been conducted uses a recursive learning base to deal with forecasting atmospheric variables. To complete sequence learning, the LSTM module is utilized, with considerations of better performance than other recursive layers as shown in Section 4.2. Meanwhile, the CNN module is used to capture the spatial pattern that represents the propagation of PM2.5 in a timestamp. The proposed model overcomes the problem of lack of heat map depiction by capturing short and long-short spatiotemporal cues at local and global levels via 3D convolution and LSTM modules. As spatiotemporal dataset holds multifaceted characteristics *e.g.,* geographical position, temporal direction, temporal speed, change in direction and distance travelled information of moving atmospheric objects such as PM2.5. By analyzing these spatiotemporal characteristics, the PM2.5 behavioral patterns can be inferred more precisely than by using conventional approach.

The proposed model is an improved version of the simple recursive network in Figure 3. The basic model is shown in Figure 5. This model is inspired by autoencoder, but it uses a Conv-LSTM2D layer instead of the standard convolutional layer. Then, the model is upgraded using the CNN-LSTM approach. Each spatial data from sensor nodes will be processed at one time using the CNN module. The CNN is applied recursively to all inputs $\theta_{m,n(t, t-1, t-2, ..., t-n)}$, *i.e.,* the heat map of PM2.5 concentration. Then, each spatial feature is processed using the ConvLSTM2D module to extract temporal features from the dataset. Finally, based on empirical evidence, the extensive experiments on 24 forecast sequences are conducted and compared with the ground truth dataset.

Figure 7 shows the CRNN network which illustrates in detail the proposed model. The k kernel size, step rate s, and output dimensions are expressed in the following order and include brackets (k, s) $[b, t, M, N, D]$ on each layer. Where, b, t, M, N, and D represent the batch size, the number of samples taken by the Conv-LSTM module to capture temporal information, the height and width of the frame, and the number of output feature maps. The network has 9 layers with 78,261 trainable parameters that integrate four main components, *i.e.,* CNN encoder, recursive layer, CNN decoder, and recursive predictor. It maintains a constant number of filters (16) in each layer (except the penultimate layer, which yields 20 feature maps) and the kernel size $k = 3$. Therefore, the spatial dimensions of the feature map are maintained similar from input to output. Thus, the encoder final layer generates a feature map that has a spatial dimension of 40 × 40. Because the network input layer accepts a frame with 40 x 40 spatial dimensions, so the preprocessing data should be done before. To achieve a precisely decoded feature map, there are four sequentially connected mini-decoder blocks. Where, each block group Conv2D and ConvLSTM2D, sequences. Therefore, the last layer of the decoder generates a feature map with the same spatial dimensions as the network input. The final classifier module consists of Batch Normalization (BN) and 3D

Convolutional layer with Sigmoid function as the classifier. Furthermore, this feature map is merged with the raw geographical map using the concatenate operator. Thus, the output of this model is a probability map of the next frame which is estimated based on the observed frames.

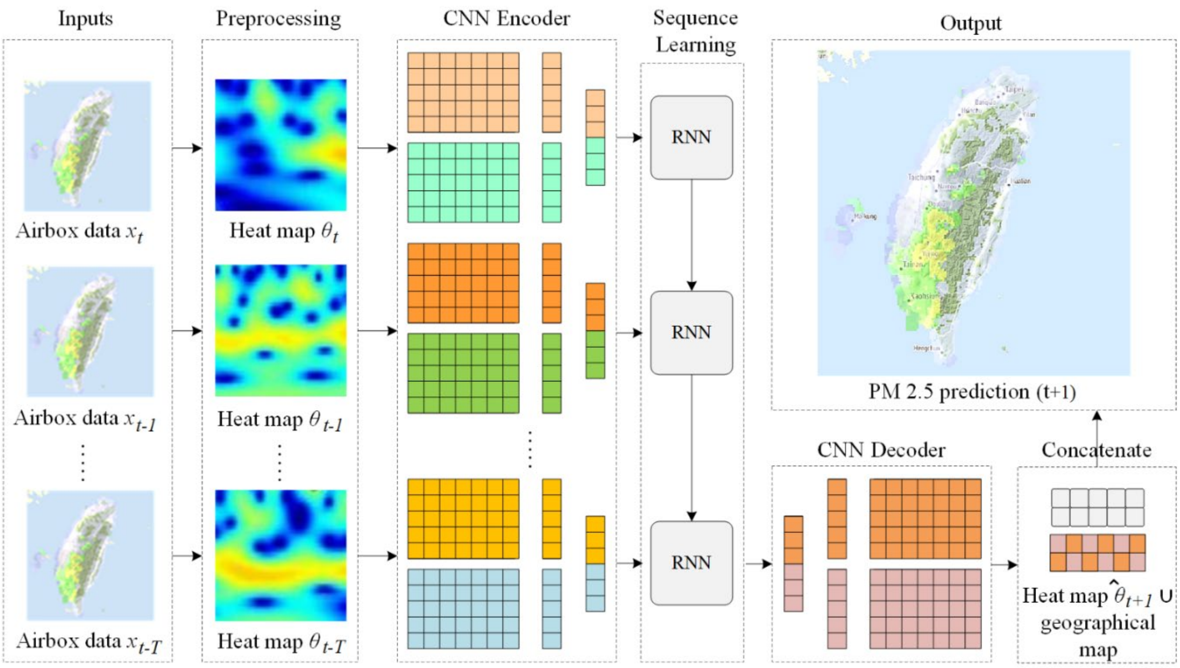

Figure 5. Design of Predictor Framework based on CNN-RNN, consist of input, preprocessing, CNN encoder, RNN-based learning, and CNN decoder

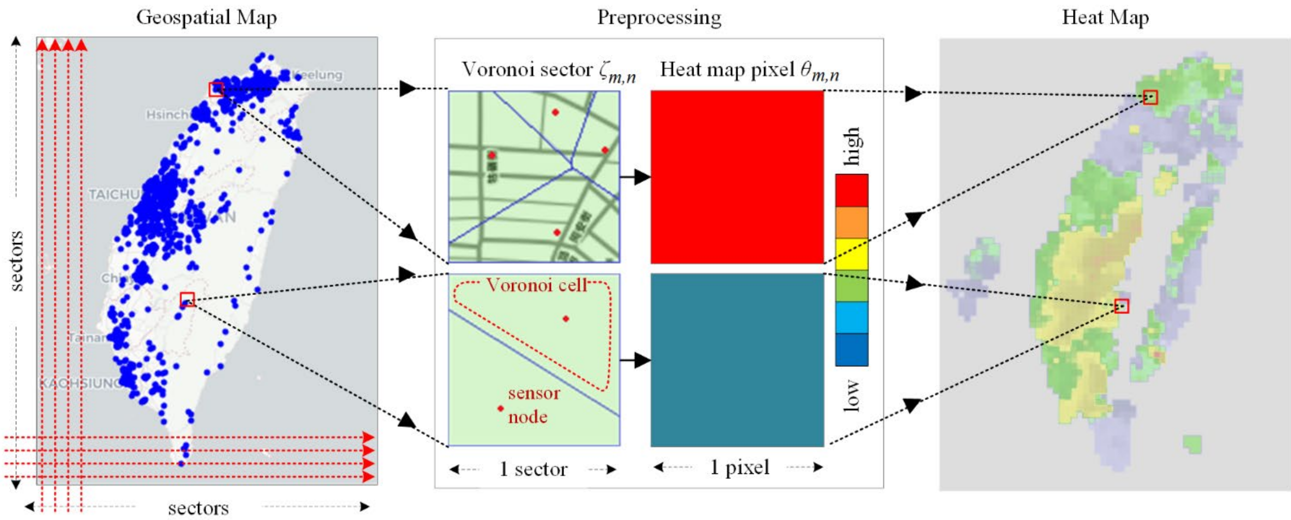

Figure 6. The preprocessing procedure involves the conversion process which transfers the geospatial position of sensor nodes into Voronoi map. Then, the heat map is generated by using mean value operation for all nodes in each Voronoi section

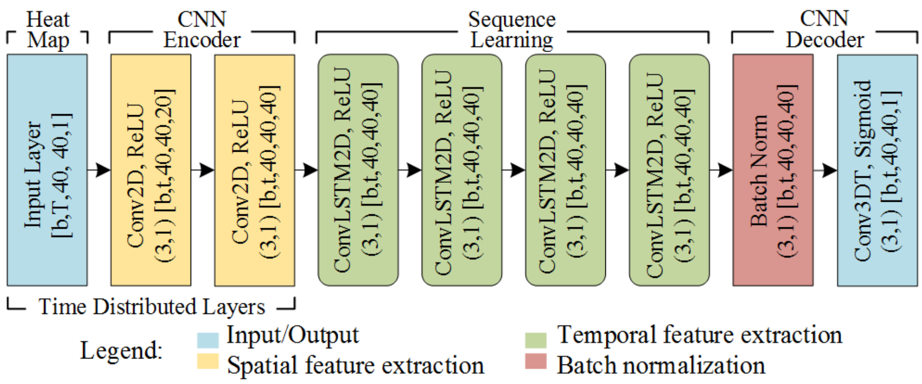

Figure 7. A layer-wise schematic of the proposed deep learning model. It exploits autoencoder-like with CRNN strategy

4.3.3 Training Strategy

Experiments were carried out on the sequence of the dataset 24 hours back to determine the forecast results for the next 24 hours. This means, the number of forecasts compared with the reference dataset is 1:1. This ratio is kept not too small so that the number of neurons required in the modeling is not too large. While the ratio is also not too close to zero so that the model has sufficient references to produce good forecasting accuracy. This approach is more precise than the random selection of frames to solve the data sequence prediction case. Meanwhile, the model is trained on NVidia Tesla 32 GB working in desktop DGX-1. On average, the training takes about 20 to 30 minutes depending on the properties of the dataset. The model is trained individually on each data set with the Adadelta optimizer described in Equation (2). Learning rate is set to 0.0002 with a scheduler reducing learning speed by factor of 0.8.

$$E = \frac{-1}{n}\sum_{n=1}^{N}\left[p_n \log \hat{p}_n + (1-p_n)\log(1-p_n)\right], \tag{2}$$

where it takes two inputs, *i.e.,* first one is the output from the final layer of the network with dimension of $b \times T \times M \times N \times D$, which maps the pixel probabilities $\hat{p}_n = \gamma(x_n) \in [0,1]$ using Sigmoid classifier γ. And another one is target $p_n \in [0,1]$ with the same dimension as the first one. p_n is the normalized ground truth map.

5 Experiments and Results

This experiment is carried out by involving the training and validation procedures. To produce optimal measurements, it is necessary to tune the model's parameters. Then, the results will be presented, followed by a comparative analysis of several approaches and their effectiveness in generating a PM2.5 prediction map. Finally, the results are visualized on a sequence map.

5.1 Fine Tuning the Proposed Model

Several parameters of the CRNN model are tuned, consisting of the number of neurons per layer, epoch, and batch size. The tuning procedure needs to be carried out to generate the most efficient number of parameters so that, the training process could be finished in the shortest possible time. In the tuning process, the optimal value of the hyperparameters is determined. The more the number of neurons, the slower the training is completed. Therefore, this larger number of neurons does not effectively contribute too much to the training accuracy. Too many epochs and batches result in more time to complete the training process even though, the accuracy slightly improves. In the experiments, the batch size = 20 and the epoch = 500. A comparative result is provided to compare the performance of NN [20], LSTM [30], CNN [32], Convolutional + LSTM (ConvLSTM) [33], and the proposed CRNN model with a total parameter of 8,459, 16,633, 43,861, 46,281, and 78,261 respectively. The simulation notes that the execution time per epoch at the training stage for the NN, LSTM, CNN, ConvLSTM, and CRNN model is 2 s, 4 s, 13 s, and 15 s. Although the learning process becomes more complex, however, at the implementation stage, the computation time will almost be the same for each NN-based model because only the inference engine is used. In addition, only the inference engine will be used more dominantly to make predictions.

5.2 Evaluation

For evaluation stage, standard performance based on Root Mean Square Error (RMSE) is calculated. The RMSE evaluates the similarity between the heat map predictions $\hat{\theta}_{t+1}$ and the ground truth θ_{t+1}. This standard is a measure of the average relative error for each pixel. The lowest the ratio, the better the performance of the prediction system. Let $\theta = \theta_{1,1}, \theta_{1,2}, \theta_{1,3}, \ldots, \theta_{m,n}$ be the ground truth data and $\hat{\theta} = \theta_{1,1}, \hat{\theta}_{1,2}, \theta_{1,3}, \ldots, \hat{\theta}_{m,n}$ be the predicted heat map. Then, the RMSE can be defined as:

$$NRMSE = \frac{1}{\bar{\theta}}\sqrt{\frac{1}{MN}\sum_{m}^{M}\sum_{n=1}^{N}\left(\theta_{m,n} - \hat{\theta}_{m,n}\right)^2}, \tag{3}$$

where $\bar{\theta}$, M and N are mean, max height, and max weight of the original heat map, respectively.

The robustness of the models is also tested and compared for each model. When implemented with real conditions, there are possibilities of errors in the data measurement process. In this condition, with the increasing number of nodes connected in the WSN, the chances of sensor damage, data transmission errors, and sparse data could occur. The robustness is evaluated by adding noise with the standard deviation σ. The greater the σ, the greater the damage to the dataset. An approach that utilizes spatiotemporal feature extraction could be an alternative to reduce the dataset fault because of noise.

5.3 Results

The test results are divided into two, namely qualitative and quantitative assessments. Quantitative assessments are obtained using the NRMSE graph as shown in Figure 8 and Figure 9. Meanwhile, quantitative assessments are carried out by visually examining the prediction results with ground truth as the target described in Figure 10.

As provided in Figure 8, all models perform good prediction proven by an accuracy value above 0.75 for the period $t+1$. In this period, the ConvLSTM generates the highest accuracy results with an NRMSE value of 0.912 followed by the proposed model with a value of 0.908. The other models generate NRMSE values of 0.87, 0.85, and 0.78 for CNN, LSTM, and NN, respectively. One-dimensional forecasting methods such as LSTM and NN cannot match the performance of 2-dimensional forecasting models, *i.e.,* CNN, ConvLSTM, and CRNN. Furthermore, the next prediction $t+2$ shows a significant performance difference between the five models tested. The one-dimensional forecasting model cannot compensate for the other three models with a significant decrease in the NRMSE value. This could occur because the weight distribution in the one-dimensional network model is very small and there are no long-term memory modules. The spatiotemporal dataset is very suitable to be extracted by using a 2-dimensional forecasting model *i.e.,* ConvLSTM. In order to more deeply observe the relationship between spatial and temporal data, Gaussian noise is added to the dataset. How strong the relationship between variables in the space and time domain can be captured and then modeled so as to improve forecasting performance.

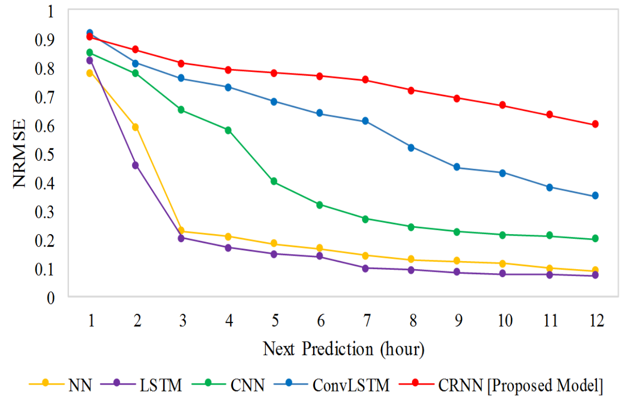

Figure 8. The RMSE of 5 predictive models involving the use of results $\hat{\theta}_{t+1}$ as input θ_t

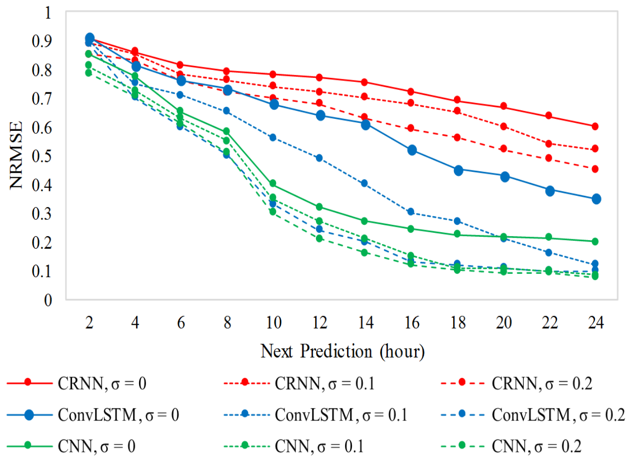

Figure 9. RMSE of the top-3 models *i.e.,* CRNN, ConvLSTM, and RNN with various of noise addition σ

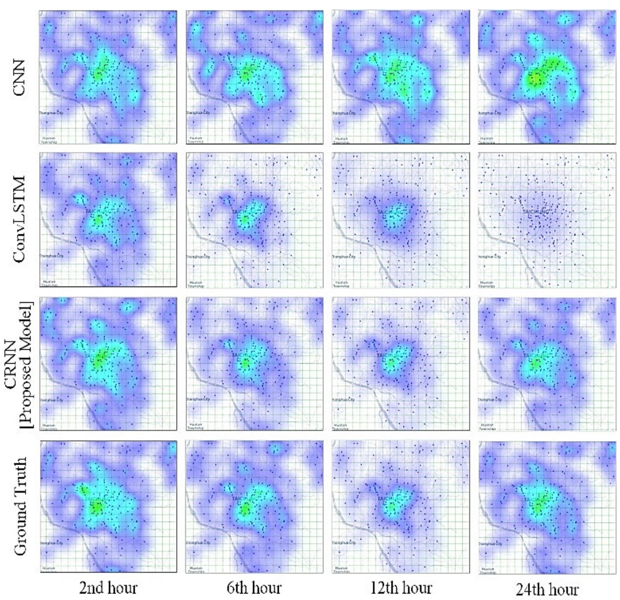

Figure 10. Propagation map of PM2.5 in Central Taiwan generated by several deep learning models

From Figure 9, it can be seen that the proposed model is able to produce the best performance compared with other 2-dimensional models when dealing with noise that occurs in the measurement system. Even when given noise $\sigma = 0.2$, the proposed model is still able to compensate for the performance of the ConvLSTM model without noise addition. The addition of noise actually worsened the performance of the other models. Finally, it is evident that the proposed model has an advantage over the conventional approaches, visually shown in Figure 10.

6 Discussion

After analyzing in our model, the LSTM model generally performs better than the GRU and RNN models in overcoming the high randomness level of PM2.5 dataset. Furthermore, it coud be used to predict the direction and rate of PM2.5 propagation. The more nodes involved, the better the forecasting performance. On a simple LSTM model, the use of more than two nodes actually makes performance decrease. Based on the tests, using only one input variable on the LSTM module, optimal performance can be achieved without sacrificing an unnecessary increase in the number of neurons. The complexity of forecasting is higher with the increasing number of variables involved. A simple recursive network has not capable to extract these patterns. The conventional approaches only capture temporal patterns. In fact, forecasting that involves massive datasets should have spatial patterns that are sometimes not recorded when using only conventional recursive models.

The proposed model provides an advantage in areas where sensor distribution is sparse. The model generates a higher resolution with a more balanced distribution by taking into account the training data from neighboring nodes. Forecasting resolution can be determined dynamically by setting the number of sectors in preprocessing. The spatiotemporal features could be recorded better using the proposed approach. The accuracy over period of forecasting could also be extended if the dataset is larger. In addition, with the increasing number of training datasets, it needs to be compensated by an increase in the number of neurons and the depth of the layer. Increasing the number of neurons will increase the training time. However, the use of CRNN has been proven to be used in PM2.5 forecasting system applications without considering other atmospheric variables. The influence of other atmospheric variables can be suppressed by learning the propagation of PM2.5 pollutants using a deep learning model so that it improves the efficiency of data processing in a massive-scale WSN.

7 Conclusions

This paper presents the model according to the idea of spatiotemporal correlation among sensory nodes, which deals with the dataset that is mined from large-scale sensor networks. By using a deep learning CRNN model, the heat maps that forecast propagation of PM2.5 concentrations over time can be generated. The proposed model adopts a combination of convolutional networks and recursive networks. The CRNN model increases the trained pattern complexity by adopting spatiotemporal dataset. Moreover, the model is powerful for learning both local and global features better than previous approaches. The network parameters are trained directly from the ground truth heat maps to generate a heat map of PM2.5 prediction. These inputs and outputs heat maps have the same format. Therefore, the proposed model does not require the retraining procedure when reconfiguring its sensory nodes *e.g.,* adding, subtracting, or replacing sensory nodes. The qualitative and quantitative results show that the proposed model could be used to make predictions without involving many atmospheric variables. In the other words, the model could minimize the use of other unnecessary variables, which are involved in pollutant propagation. Finally, the experiment results show that developing spatiotemporal feature extraction approach proposed model in this paper is a powerful solution with efficient performance. It could provide the prediction solution for the massive IoT deployment based on green communication.

Acknowledgements

This work was supported by the Ministry of Science and Technology (MOST), Taiwan, under MOST Grant No. 110-2218-E-468-001-MBK. This work was also funded by MOST under Grant No. 110-2221-E-468 -007. This work was supported by MOST under Grant No. 110-2218-E-002-044. This work was supported by Asia University, Taiwan, UMY, Indonesian, under Grant No. 107-ASIA-UMY-02.

References

[1] C.-P. Chio, W.-C. Lo, B.-J. Tsuang, C.-C. Hu, K.-C. Ku, Y.-J. Chen, H.-H. Lin, C.-C. Chan, Health impact assessment of PM2.5 from a planned coal-fired power plant in Taiwan, *Journal of the Formosan Medical Association*, Vol. 118, No. 11, pp. 1494-1503, November, 2019.

[2] A. Y. P. Wardoyo, U. P. Juswono, J. A. E. Noor, Measurements of PM2.5 motor emission concentrations and the lung damages from the exposure mice, *2016 International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM)*, Malang, Indonesia, 2016, pp. 99-103.

[3] Md. F. Khan, Y. Shirasuna, K. Hirano, S. Masunaga, Characterization of PM2.5, PM2.5–10 and PM>10 in ambient air, Yokohama, Japan, *Atmospheric Research*, Vol. 96, No. 1, pp. 159-172, April, 2010.

[4] Z. Wei, J. Peng, X. Ma, S. Qiu, S. Wang, Toward Periodicity Correlation of Roadside $PM_{2.5}$ Concentration and Traffic Volume: A Wavelet Perspective, *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 11, pp. 10439-10452, November, 2019.

[5] R. Borge, W. J. Requia, C. Yagüe, I. Jhun, P. Koutrakis, Impact of weather changes on air quality and related mortality in Spain over a 25 year period [1993–2017], *Environment International*, Vol. 133, Part B, Article No. 105272, pp. 1-14, December, 2019.

[6] N. V. A. Grace, A. Chilambuchelvan, J. Shanmugapriyan, Prediction based compression algorithm for univariate and multivariate data in wireless communication networks, *Journal of Internet Technology*, Vol. 21, No. 2, pp. 489-499, March, 2020.

[7] K. T. Putra, H.-C. Chen, Prayitno, M. R. Ogiela, C.-L. Chou, C.-E. Weng, Z.-Y. Shae, Federated Compressed Learning Edge Computing Framework with Ensuring Data Privacy for PM2.5 Prediction in Smart City Sensing Applications, *Sensors*, Vol. 21, No. 13, Article No. 4586, pp. 1-20, July, 2021.

[8] H.-C. Chen, K. T. Putra, S.-S. Tseng, C.-L. Chen, J. C.-W. Lin, A spatiotemporal data compression approach with low transmission cost and high data fidelity for an air quality monitoring system, *Future Generation Computer Systems*, Vol. 108, pp. 488-500, July, 2020.

[9] X. Li, D. Long, An improvement in accuracy and spatiotemporal continuity of the MODIS precipitable water vapor product based on a data fusion approach, *Remote Sensing of Environment*, Vol. 248, Article No. 111966, pp. 1-16, October, 2020.

[10] Edimax Technology Co., Ltd, AirBox: Air Quality Detector with PM2.5, Temperature and Humidity Sensors, *AirBox Datasheet*: *AI-1001W V2*, pp. 1-3, 2017.

[11] J. He, G. Christakos, P. Jankowski, Comparative Performance of the LUR, ANN, and BME Techniques in the Multiscale Spatiotemporal Mapping of PM2.5 Concentrations in North China, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 12, No. 6, pp. 1734-1747, June, 2019.

[12] R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, F. Li, Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering, *Expert Systems with Applications*, Vol. 169, Article No. 114513, pp. 1-15, May, 2021.

[13] S.-H. Hur, Short-term wind speed prediction using Extended Kalman filter and machine learning, *Energy Reports*, Vol. 7, pp. 1046-1054, November, 2021.

[14] B. Xu, W. Lin, S. A. Taqi, The impact of wind and non-wind factors on $PM_{2.5}$ levels, *Technological Forecasting and Social Change*, Vol. 154, Article No. 119960, pp. 1-14, May, 2020.

[15] M.-Y. Lin, Y.-X Guo, Y.-C. Chen, W.-T. Chen, L.-H. Young, K.-J. Lee, Z.-Y. Wu, P.-J. Tsai, An instantaneous spatiotemporal model for predicting traffic-related ultrafine particle concentration through mobile noise measurements, *Science of The Total Environment*, Vol. 636, pp. 1139-1148, September, 2018.

[16] Y. Zhang, Z. Huang, S. Wen, Spatiotemporal variations of in-cabin particle concentrations along public transit routes, a case study in Shenzhen, China, *Building and Environment*, Vol. 180, Article No. 107047, pp. 1-15, August, 2020.

[17] Y. Etzion, D. M. Broday, Highly resolved spatiotemporal variability of fine particle number concentrations in an urban neighborhood, *Journal of Aerosol Science*, Vol. 117, pp. 118-126, March, 2018.

[18] P. S. S. de Souza, F. P. Rubin, R. Hohemberger, T. C. Ferreto, A. F. Lorenzon, M. C. Luizelli, F. D. Rossi, Detecting abnormal sensors via machine learning: An IoT farming WSN-based architecture case study, *Measurement*, Vol. 164, Article No. 108042, pp. 1-7, November, 2020.

[19] W. Wang, Y. Guo, Air Pollution PM2.5 Data Analysis in Los Angeles Long Beach with Seasonal ARIMA Model, *2009 International Conference on Energy and*

Environment Technology (ICEET 2009), Guilin, China, 2009, pp. 7-10.

[20] F. D. Syahfitra, R. Syahputra, K. T. Putra, Implementation of Backpropagation Artificial Neural Network as a Forecasting System of Power Transformer Peak Load at Bumiayu Substation, *Journal of Electrical Technology UMY*, Vol. 1, No. 3, pp. 118-125, September, 2017.

[21] J. Di, Investigation on the traffic flow based on wireless sensor network technologies combined with FA-BPNN models, *Journal of Internet Technology*, Vol. 20, No. 2, pp. 589-597, March, 2019.

[22] P. Du, J. Wang, Y. Hao, T. Niu, W. Yang, A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2.5 and PM10 forecasting, *Applied Soft Computing*, Vol. 96, Article No. 106620, pp. 1-20, November, 2020.

[23] Y. T. Tsai, Y. R. Zeng, Y. S. Chang, Air pollution forecasting using RNN with LSTM, *IEEE 16th International Conference on Dependable, Autonomic and Secure Computing*, Athens, Greece, 2018, pp. 1074-1079.

[24] Q. Chen, W. Wang, X. Huang, H. N. Liang, Attention-based recurrent neural network for traffic flow prediction, *Journal of Internet Technology*, Vol. 21, No. 3, pp. 831-839, May, 2020.

[25] S. Song, J. C. K. Lam, Y. Han, V. O. K. Li, ResNet-LSTM for Real-Time PM2.5 and PM_{10} Estimation Using Sequential Smartphone Images, *IEEE Access*, Vol. 8, pp. 220069-220082, December, 2020.

[26] C. Li, J. Wang, M. Li, Spatial-temporal compressive sensing for cross-layer optimization data transmission in wireless sensor networks, *Journal of Internet Technology*, Vol. 19, No. 5, pp. 1457-1463, September, 2018.

[27] O. Barron, M. Raison, G. Gaudet, S. Achiche, Recurrent Neural Network for electromyographic gesture recognition in transhumeral amputees, *Applied Soft Computing*, Vol. 96, Article No. 106616, pp. 1-9, November, 2020.

[28] J. Song, G. Xue, Y. Ma, H. Li, Y. Pan, Z. Hao, An Indoor Temperature Prediction Framework Based on Hierarchical Attention Gated Recurrent Unit Model for Energy Efficient Buildings, *IEEE Access*, Vol. 7, pp. 157268-157283, October, 2019.

[29] W. Qiao, W. Tian, Y. Tian, Q. Yang, Y. Wang, J. Zhang, The Forecasting of PM2.5 Using a Hybrid Model Based on Wavelet Transform and an Improved Deep Learning Algorithm, *IEEE Access*, Vol. 7, pp. 142814-142825, September, 2019.

[30] C. T. Yang, L. Y. Lin, Y. T. Tsan, P. Y. Liu, W. C. Chan, The Implementation of a Real-time Monitoring and Prediction System of PM2.5 and Influenza-Like Illness Using Deep Learning, *Journal of Internet Technology*, Vol. 20, No. 7, pp. 2237-2245, December, 2019.

[31] S. Vimal, Y. H. Robinson, S. Kadry, H. V. Long, Y. Nam, IoT based smart health monitoring with CNN using edge computing, *Journal of Internet Technology*, Vol. 22, No. 1, pp. 173-185, January, 2021.

[32] B. Pu, Y. Liu, N. Zhu, K. Li, K. Li, ED-ACNN: Novel attention convolutional neural network based on encoder–decoder framework for human traffic prediction, *Applied Soft Computing*, Vol. 97, Article No. 106688, pp. 1-15, December, 2020.

[33] T. Li, M. Hua, X. Wu, A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5), *IEEE Access*, Vol. 8, pp. 26933-26940, February, 2020.

[34] W. Li, L. Zhu, Y. Shi, K. Guo, E. Cambria, User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models, *Applied Soft Computing*, Vol. 94, Article No. 106435, pp. 1-11, September, 2020.

[35] W. Li, Y. Xu, G. Wang, Multi-target stance detection based on GRU-PWV-CNN network model, *Journal of Internet Technology*, Vol. 22, No. 3, pp. 593-603, May, 2021.

[36] F. Aurenhammer, Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure, *ACM Computing Surveys*, Vol. 23, No. 3. pp. 345-405, September, 1991.

Biographies

Hsing-Chung Chen received the Ph.D. degree in Electronic Engineering from National Chung Cheng University, Taiwan, 2007. Since Aug. 2019–present, he has been Distinguished Full Professor of Computer Science and Information Engineering, Asia University, Taiwan. His research interests include Information and Communication Security, Blockchain Technology and Security, AIoT, 5G and Wireless Networks, Medical and Bio-information Signal Image Processing, Artificial Intelligence and Applied Cryptography.

Karisma Trinanda Putra currently a Ph.D. student at the Department of Computer Science and Information Engineering at Asia University, Taiwan. He is a lecturer in Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Indonesia. His research interests include Data Mining, Machine Learning, and Sensor Networks.

Chien-Erh Weng received the Ph.D. degree in Electrical Engineering from the National Chung Cheng University, Chiayi, Taiwan, 2007. Recently, he is the Professor at National Kaohsiung University of Science and Technology. He is also the Vice Dean of R&D office. His research interests include Signal Processing, Data Analysis, 5G/B5G Communication System, Machine Learning and Automatic Identification System.

Jerry Chun-Wei Lin received his Ph.D. from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan in 2010. He is currently a full Professor with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway. His research interests include Data Mining, Soft Computing, Artificial Intelligence/Machine Learning, and Privacy Preserving and Security Technologies.