Neural Computing and Applications (2021) 33:14199–14229
https://doi.org/10.1007/s00521-021-06067-7

ORIGINAL ARTICLE

# Intelligent forecaster of concentrations (PM2.5, PM10, NO2, CO, O3, SO2) caused air pollution (IFCsAP)

Samaher Al-Janabi[1] · Ayad Alkaim[2] · Ehab Al-Janabi[3] · Aseel Aljeboree[2] · M. Mustafa[1]

Received: 18 September 2020 / Accepted: 19 April 2021 / Published online: 20 May 2021


## Abstract

Upgrading health reality is the responsibility of all, it is necessary to think about the design of a smart system based on modern technologies to reduce the time and effort exerted on the competent authorities in both health and environmental sectors furthermore making their work environment smart and easy to enable the creativity and innovation as well as to reduce the material costs granted by state of this case "environment." The process to find a solution for the problem of a triangle as shown in Figure (1) with contradictory heads is a very important and difficult issue in the field of health and environment, these are: (to optimize time utilization, and minimize the human errors, that accompany this human effort as much as possible, and to reduce material costs). Therefore, the idea of Internet technology and the Intelligent Big Data Analysis was developed to design an integrate electronic system of hardware to be developed in different and specific locations to collect information on concentrations that cause air pollution. So, it was invested an idea of Internet of things technology and intelligent data analysis ("Internet of things" and "Intelligent Data Analysis") for the construction of an integrated system of Hardware entities and Software entities placed. The aim of this work is to build a programmable system capable of predicting the pollutant concentrations within the next 48 h called intelligent forecaster of concentrations caused air pollution (IFCsAP) and making the machine the primary source of information after these concentrations are collected and stored in real time. On this basis, we will rely on modern technologies to reach this goal. The proposed design is highly efficient, cost-effective and easy to use and can be deployed in all places (environment with concentrations of air pollution). The main objective of the proposed system is to issue periodic reports (within 48 h of the future) based on the information input from different stations in real time. These reports are issued based on the readings of sensors planted at each station. Each sensor has a measurement of one or more concentrations that cause air pollution. Designed system consists of three basic phases: the construction phase of an integrated electronic circuit consisting of several devices (Modern, LoRa, Waspmate Platform, Arduino, Five sensors).

**Keywords** IFCsAP · DNS-PSO · DLSTM · PM2.5 · PM10 · NO2 · CO · O3 · SO2 · Air quality index

**Abbreviations**

| | |
|---|---|
| DLSTM | Developed long short–term memory |
| LSTM | Long short-term memory |
| PSO | Particle swarm optimization |
| SMAPE | Symmetric mean absolute percentage error |
| PM2.5 | Particulate matter that has a diameter of less than 2.5 μm |
| PM10 | Particulate matter 10 μm or less in diameter |

✉ Samaher Al-Janabi
samaher@itnet.uobabylon.edu.iq

Ayad Alkaim
alkaim@iftc.uni-hannover.de

Ehab Al-Janabi
Eng.aljanabi.ehab@gmail.com

Aseel Aljeboree
aselo91@gmail.com

M. Mustafa
musmus55@gmail.com

[1] Department of Computer Science, Faculty of Science for Women (SCIW), University of Babylon, Babylon, Iraq

[2] Department of Chemistry Science, Faculty of Science for Women (SCIW), University of Babylon, Babylon, Iraq

[3] Babylon Electricity Distribution Branch, Electricity Distribution Company for the Middle, Ministry of Electricity, Babylon, Iraq

| | |
|---|---|
| O3 | Ozone is the unstable triatomic form of oxygen |
| Sox | Sulfur oxides |
| CO | Carbon monoxide |
| NOx | Nitrogen oxides |
| $\odot$ | Is the element-wise product or Hadamard product |
| $\otimes$ | Outer products will be represented |
| $\sigma$ | Represents the sigmoid function |
| at | Input activation |
| it | Input gate |
| ft | Forget gate |
| ot | Output gate |
| Statet | Internal state |
| $\mathrm{Out}_t$ | Output |
| $W$ | The weights of the input |
| $U$ | The weights of recurrent connections |
| $V_i^t$ | Particle velocity $i$ in swarm in dimension $j$ and frequency $t$. |
| $X_i^t$ | The location of the particle $i$ in a swarm in dimension $j$ and frequency $t$. |
| $c_1$ | Acceleration factor related to Pbest. |
| $c_2$ | Acceleration factor related to gbest. |
| $r_1^t$, $r_2^t$ | Random number between 0 and 1. |
| $t$ | Number of occurrences specified by type of problem. |
| $G_{\mathrm{best},i}^t$ | Gbest position of swarm |
| $P_{\mathrm{best},i}^t$ | Pbest position of particle |

## 1 Introduction

Air pollution is one of the most important challenges facing the world today as a result of the development of technology [1, 2], where it can be defined from several aspects in terms of pathogenesis. Pollution due to the presence of living or invisible organisms, such as bacteria and fungi, in the environment such as water, air or soil (Air pollution as chemical is the imbalance of the ecosystem by chemical effects, and these pollutants can be in the form of solid particles or liquid droplets or gases), from the scientific point of view (a change in the harmonic movement between the components of the ecosystem that paralyzes the efficiency of this system and loses its ability to perform its natural role in the self-disposal of pollutants). This research deals with intelligent predictive design to address this phenomenon [3]. There are different types of error measurement, including: Root Mean Square Error (RMSE) measures how much error there is between two data sets. In other words, it compares a predicted value and an observed or known value. It's also known as Root Mean Square Deviation and is one of the most widely used statistics in GIS [4]. $\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(F_i - A_i)^2}{n}}$ Where: F = forecasts (expected values or unknown results), A = observed values (known results). n = sample size. Other measures known, cross-entropy loss is another loss function mostly used in regression and classification problems. Cross-entropy loss [5] is given by $H(A) = -\sum_i F_i \log(A_i)$ where $y_i^-$ is the target label, and $y_i$ is the output of the classifier. Cross-entropy loss function is used when the output is a probability distribution, and thus it is preferred [6]. While, symmetric mean absolute percentage error (SMAPE) is an accuracy measure based on percentage (or relative) errors. It is usually defined as [7]: $\mathrm{SAMPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{|A_t + F_t|/2}$ where At is the actual value and Ft is the forecast value. The absolute difference between At and Ft is divided by half the sum of absolute values of the actual value At and the forecast value Ft. The value of this calculation is summed for every fitted point t and divided again by the number of fitted point's n. If actual value and forecast value are both 0, we will set SMAPE score 0, too. This paper used SMAPE to evaluate the quality of a prediction, by comparing predicted to observed values.

Forecasting is one of taking decision process to find estimates values for the future based on past data [8]. There are three type of prediction [9]: First, perspective predication model indicates the task of developing a model that is aimed to predict the target's value as a work of the informative variable and the main aim of these tasks is predicting the value regarding a specific attribute according to the other attribute values. Second, Traditional Prediction: During the first half of the twentieth century, many methods used to extrapolate the future were used for decision-making. They are part of the planning process, at the same time, they have succeeded in helping planners predict and make rational decisions about the future, it is considered a traditional means of dealing with the future when compared to modern methods and techniques in this field. Traditional methods include: Method of prediction by guessing: This method depends on the intuitive way used by the individual in assessing some aspects of the future. But such predictions may fail more than success (Fig. 1).

Deep techniques are set of multi-levels learning techniques derivative from automated learning [10, 11]. A field in which the computer tests algorithms and programs by learning to improve and develop it by itself. Modern computer vision, speech recognition programs and future prediction are all the product of deep learning [12, 13]. The need for this method increases with the emergence of the concept of large data. Because of its ability to deal with these data, so the computer needs preliminary data to understand the relationship between objects, if we can say that it is a set of algorithms that allow the device to learn

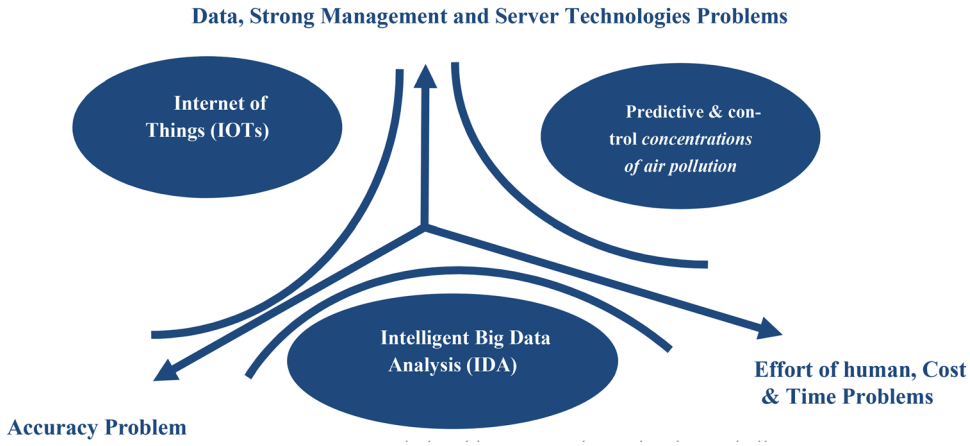


**Fig. 1** Relationships among the main three challenges

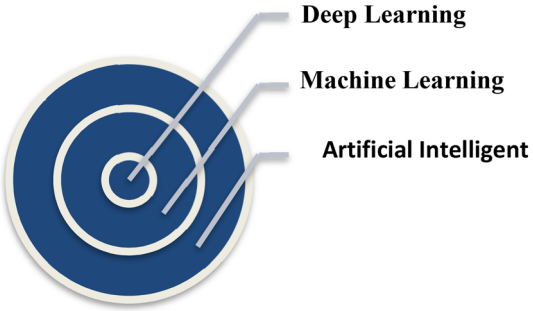


**Fig. 2** Relationship among AI, machine learning and deep learning

from itself and events, this makes the device learns and then develops itself through the neural classes [14, 15]. The greater the number of neural classes, the greater the performance of the device. This is characterized by deep instruction in teaching the device on other techniques that have a certain level of learning, injury to develop with the increase in the volume of data. To ensure the quality of automated learning through deep learning, you must provide and enter as many data as possible and to illustrate the relationship among these terms can be conceived in the form of concentric circles as explain in Fig. 2.

In this paper, we will present new forecaster through synthesis between tow techniques LSTM and PSO after develop one of it through build DSN-PSO to enhance the performance of other LSTM to satisfy the following: highly efficient, least cost and easy to use. Before build the forecaster, we design electric circuit consisting of several devices (LoRa, Waspmate Platform, Five sensors).

LoRa is modulation technique which allows sending data at extremely low data-rates to extremely long ranges for more detail see [16].

'Waspmote platform is an architecture available as open access allow by connect devise and sensors platform, for more detail see [16, 17]. A sensor is a device, module, machine, or subsystem used in many applications to read the data for specific event or change in specific environment in the real times, for more detail see [18].

In this work, we deal with five types of sensors are Grove—Laser PM2.5 Sensor (HM3301) to measure PM2.5, PM10, MQ-7 to measure carbon monoxide (CO), MQ131 to measure Ozone, and NO2 sensor to measure nitrogen dioxide (NO2) used to collect data in real-time Fig. 3 showing the electrical circuit connects the main parts of station.

The main points attempt the achieve in this work:

- Increase accuracy in knowing the percentage of air pollution in the coming days to take precautionary measures against the risks of such pollution and try to reduce it.
- This integrated system is part of the electronic management and chemical safety of laboratories.
- Apply decisions from health and environmental communities and avoid early pollution risks by educating people.
- The system provides us with important statistical information in raw form that can contribute to the treatment of sources that cause pollution of air produced by human effort such as factories, houses or produced by nature such as burning forests and volcanoes and others and guidance.
- The system is inexpensive and therefore does not burden the Ministries of Health and Environment.
- To achieve the innovative method of safety of personnel working in laboratories that deal with these chemicals and comply with the requirements of UNESCO for the achievement of chemical safety and safety conditions.

The sensor is a device that works to detect the physical or chemical ambient state, some measure temperature, some measure pressure, some measure gases, and some measure air quality. It converts the signals incident upon it into electrical impulses that can be measured or counted by

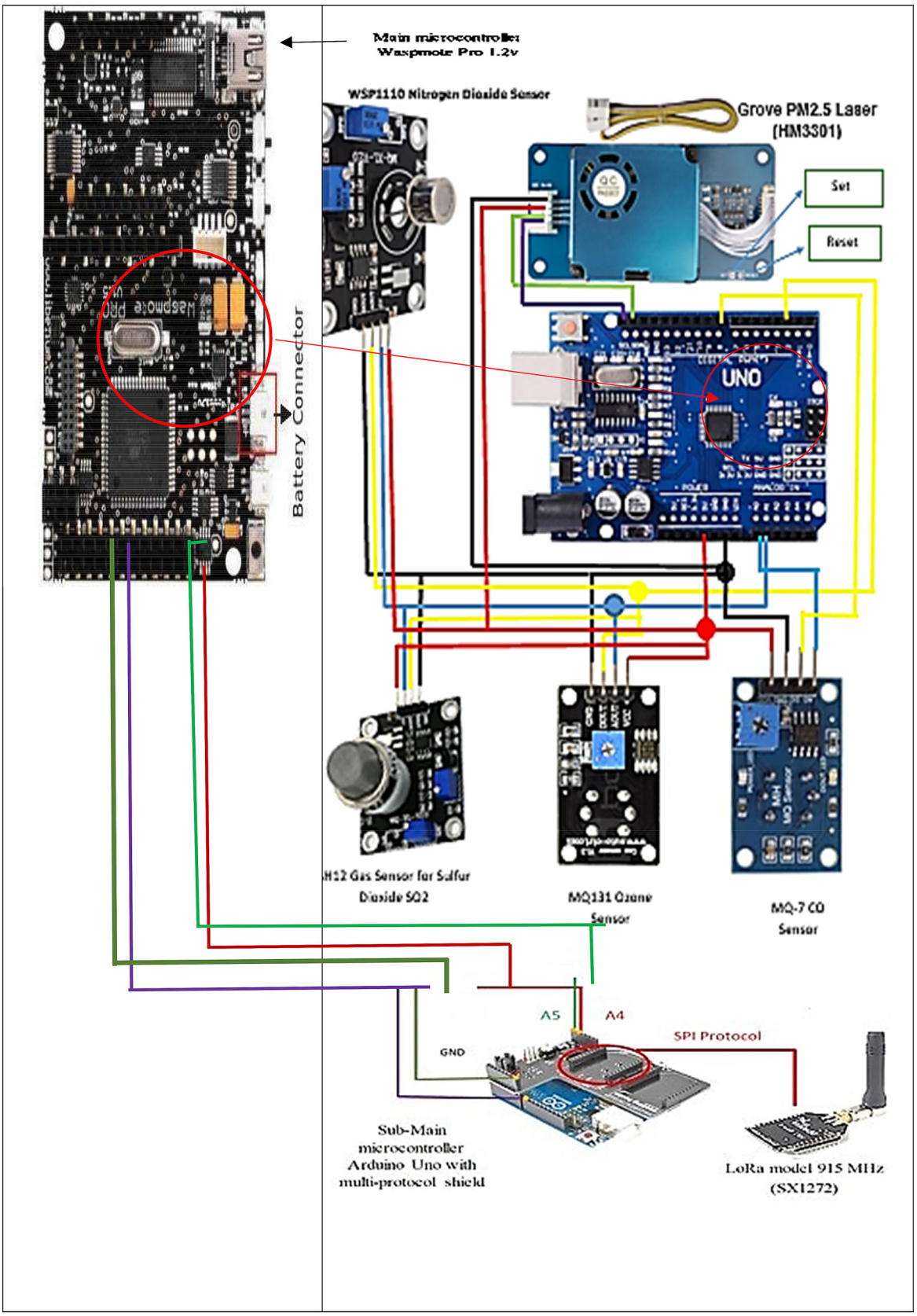


**Fig. 3** Connect the main sensors in design station through LoRa model

a device such as a computer [27]. In other words, the sensor is a device, module, or subsystem that aims to detect events or changes in its environment and send information to other electronics, often a computer processor. The sensor is always used with other electronic devices.

There are many sensors to measure the concentrations that cause air pollution, but in this paper, we will focus on the sensors specific to our work:

### 1.1 Sensor–Grove PM2.5 Laser (HM3301)

It is a new generation of laser dust detection sensor, which is used for continuous and real detection of dust in the air. It is used to measure PM2.5 and PM10 concentrations.

The main features of this sensor:

- High sensitivity to dust particles 0.3 μm or greater.
- Continuous detection of dust concentration in the air in real time.
- Based on laser light scattering technology, readings are accurate, stable and consistent.
- Low noise.
- Energy consumption is very low.

### 1.2 Sensor–MQ-7

The MQ-7 gas sensor has high sensitivity to carbon monoxide. The sensor can be used to detect different gases that contain carbon dioxide, so it is low in cost and suitable for different applications.

The main features of this sensor:

- High sensitivity to combustible gas (CO) in a wide range.
- Stable performance, long life, and low cost.
- Simple drive circuit.

### 1.3 Sensor-MQ131

The MQ131 gas sensor is highly sensitive to ozone.

The main features of this sensor:

- Good sensitivity to ozone in a wide range of gases.
- Long life and low cost.
- Simple drive circuit.

### 1.4 Sensor–WSP1110 nitrogen dioxide sensor

Low-cost electrochemical nitrogen dioxide sensors provide exciting new opportunities for rapid and distributed outdoor air pollution measurements. This type of sensor is stable, long lasting, requires little energy, and is capable of accurately measuring parts per billion (parts per billion).

The main features of this sensor:

- High sensitivity, stable performance and long-life time
- Small in size and light in weight
- 5 V voltage, low consumption
- Quick response reset function, simple drive circuit
- Long-term stability (50 ppm overload).

### 1.5 Sensor–SO2

SO2 sensor is designed to measure sulfur dioxide for applications in: air quality monitoring, industrial safety and air purification monitoring.

The main features of this sensor:

- Small in size with low profile (15 × 15 × 3 mm).
- Long life (10 years life expectancy).
- Fast response (15 s typical).

## 2 Related works

The issue of air quality prediction is one of the critical topics related to human lives and health. The aim of the work presented herein is to develop a new method for such prediction based on the huge amount of data that is available and operating on data series. This section first reviews previous studies by researchers in this area and compares them based on the database used in each case, the methods applied to assess the results, the advantages of each method, and its limitations.

Li et al. [19] used a long short-term memory extended (LSTME) neural network model with combined spatial–temporal links to predict concentrations of air pollutants. In that approach, the LSTM layers automatically extract potential intrinsic properties from historical air pollutant and accompanying data, while meteorological data and timestamp data are also incorporated into the proposed model to improve its performance. The technique was evaluated using three measures (RMSE, MAE, and MAPE) and compared with the STANN, ARMA, and SVR models. The work presented herein is similar in its use of the LSTM approach as part of a recurrent neural network structure but differs in its use of another evaluation measure.

Lifeng et al. [20] reported that the best predictions of air quality could be obtained using the GM model (1.1) with fractional order accumulation, i.e., FGM (1.1), to find the expected average annual concentrations of PM2.5, PM10, SO2, NO2, 8-h O3, and O-24 h. The measure used in that work was the MAPE. Application of the FGM (1.1) method

resulted in much better performance compared with the traditional GM model (1.1), revealing that the average annual concentrations of PM2.5, PM10, SO2, NO2, O8–O3, and O3 24-h will decrease from 2017 to 2020. That work presented herein is similar in that it predicts the concentration of air pollutants and finds ways to address them, but differs in its use of the LSTM method for the predictions.

Wen et al. [21] combined a convolutional neural network (CNN) and LSTM neural network (NN), as well as meteorological and aerosol data, to refine the prediction performance of the model. Data collected from 1233 air quality monitoring stations in Beijing and the whole of China were used to verify the effectiveness of the proposed model (C-LSTME). The results showed that the model achieved better performance than state-of-the-art technologies for predictions over different durations at various regional and environmental scales. The technique was evaluated using three measures (RMSE, MAE, and MAPE). In comparison, the LSTM approach is also applied in a RNN in this work, but after having identified the best structure for the network. In addition, another evaluation measure is used herein.

Shang et al. [22] described a prediction method based on a classification and regression tree (CART) approach in combination with the ensemble extreme learning machine (EELM) method. Subgroups were created by dividing the datasets using a shallow hierarchy tree through the CART approach. At each node of the tree, EEL models were constructed using the training samples of the node, to minimize the verification errors sequentially in all of the subtrees of each tree by identifying the number of hidden intestines, where each node is considered to be a root. Finally, the EEL models for each path to a leaf are compared with the root of each leaf, selecting only the path with the smallest error to check the leaf. The measures used in that work were the RMSE and MAPE. This experimental measurement results revealed that such a method can address the issue of global–local duplication of the prediction method at each leaf and that the combined CART–EELM approach worked better than the random forest (RF), v-(SVR), and EELM models, while also showing superior performance compared with EELM or k-means EELM seasonal. The work presented herein is similar in that it uses the same set of six air pollution indexes (PM2.5, O3, PM10, SO2, NO2, CO) but differs in terms of the mechanism applied to reduce air pollutants, applying the RNN method.

Li et al. [23] applied a new air quality forecasting method and proposed a new positive analysis mechanism that includes complex analysis, improved prediction units, data pretreatment, and air quality control problems. The system analyzes the original series using an entropy model and a data processing process. The multiobjective multiverse optimization (MOMVO) algorithm is used to achieve the required performance, revealing that the least-squares (LS)SVM achieved the best accuracy in addition to stable predictions. Three measures were used for the evaluation in that work, viz. RMSE, MAE, and MAPE. The results of the application of the proposed method to the dataset revealed good performance for the analysis and control of air quality, in addition to the approximation of values with high precision. The work presented herein uses the same evaluation measures but differs in its use of the LSTM approach in the RNN after identifying the best structure for the network.

Kim et.al. [24] aim to build annual-average integrated empirical geographic (IEG) regression models for the contiguous USA for six criteria pollutants during 1979–2015; explore systematically the impact on model performance of the number of variables selected for inclusion in a model; and provide publicly available model predictions. We compute annual-average concentrations from regulatory monitoring data for PM10, PM2.5, NO2, SO2, CO, and ozone at all monitoring sites for 1979–2015.

## 3 Building IFCsAP

The model presents in this paper consist of two phases, the first including build the station as electrical circuit to collect the data related to six concentrations in real time and saved it on the master computer to preparing and processing in next phase. The second phase focuses on processing dataset after splitting it based on station identifier, the processing phase pass on many levels of learning to product forecaster can deal with hug/big dataset. All the actives of this researcher summarization in Fig. 5 while the algorithm of IFCsAP model described in main algorithm. To making the model more understanding, we explain the first phase on it in Fig. 3 while the second phase in Fig. 4. The main constructions used.

- PM2.5: 10 μg/m$^3$ (average allowable value per year), 25 μg/m$^3$ (average allowable value in 24 h).
- PM10: 20 μg/m$^3$ (average allowable value per year), 50 μg/m$^3$ (average allowable value per year).
- o3: 100 μg/m$^3$ (average allowable value in eight hours). The recommended maximum value, previously set at 120 μg/in eight hours, has been reduced to 100 μg/m$^3$ based on recent findings of relationships between daily mortality and ozone levels in locations where the concentration of the substance is less than 120 μg/m$^3$.
- No2: 40 μg/m$^3$ (average allowable value per year), 200 μg/m$^3$ (average allowable value per hour).
- SO2: 20 μg/m$^3$ (average allowable value in twenty-four hours), 500 μg/m$^3$ (average allowable value in 10 min).

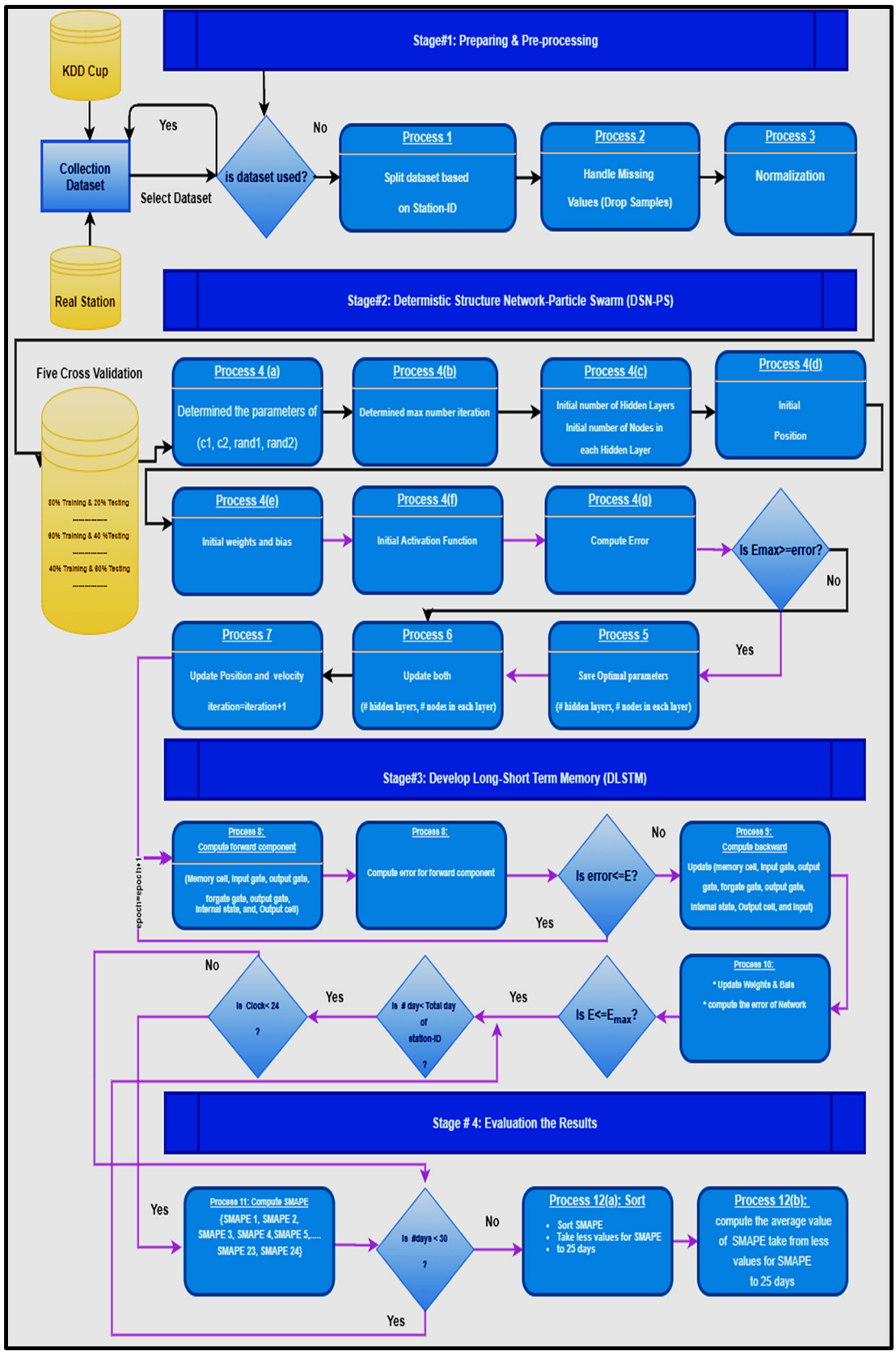


**Fig. 4** Block diagram of IFCsAP model

**Main Algorithm: IFCsAP Model**

***Input:*** *Datasets of all Stations*

***Output:*** *Determined the rate of air index quality*

1. ***Building the hardware part*** *based on six subsystems (i.e. Sensors, Processor, Arduino, Communication, Real-Time Clock, Power subsystem)*

```
2.  For each record in original dataset
3.      Divide dataset based on Stations-id
4.      For each construction in original dataset
5.          Handle missing value by drop record
6.          Make all the values in the same rage through Normalize
7.      End for
8.  End for
9.  For each station recognition through id
10.     For all samples in this station
11.         Divide samples into training and testing dataset based on 10- Cross-Validation
12.     End for
13.     For each not used training part
14.         Import DSN-PS
15.         Import DLSTM
16.     End for
17.     For each not used Testing part
18.         Investigation the stopping conditions
19.             Compare  if Emax less than max error generation
20.               Jump to 25
21.             Else
22.               Jump to 10
23.             End IF
24.     End for
25. End for
```

26. ***Import Evaluation SMAPE***

***End IFCsAP Model***

Dataset collection through two types of resources (i.e., directory web site represents by KDD cup 2018 dataset and by building station have multi-sensors to caption concentrations). That dataset needed to handle it before building the predictor as follows.

- Split the dataset for each station and save it in separated file hold the name of this station.
- After that, treatment missing values through drop each row have one or more missing values.
- Finally, apply the normalization for each column in dataset related to each station to make the value of that concentration in the range [0, 1].

### 3.1 Develop long short-term memory (DLSTM)

This paper presents how can employ PSO through build new algorithm called DSN-PSO as explained in algorithm 2 to enhance the performance of one of deep learning algorithm LSTM (i.e., for more detail see **Main steps for training LSTM–RNN** in "Appendix") through determined the structure and parameters of it. This explains with details in Algorithm 3 (Table 1).

**Table 1** The parameters utilize in models

| Parameter | Value |
|---|---|
| Learning coefficients: c1, c2 | Uniformly distributed between [0, 4] |
| Maximum number of iterations | 150 |
| Number of hidden layers | Within the range [1, 10] |
| Number of neurons in each layer | Within the range [1, 257] |
| Number of weights in the memory cell and each gate | Based on the number of nodes in each hidden layer |
| Number of bias in the memory cell and each gate | Based on the number of nodes in each hidden layer |
| Particle dimensions | Represents the number of hidden layers and the number of neurons in each layer |
| rand1, rand2 | Random numbers that are in the range [0, 1] |
| Calculate velocity and position of hidden layers | $vh = vh + c1 * \text{rand} * (\text{phBest} - h) + c2 * \text{rand} * (\text{ghBest} - h)$(1)<br>$h = h + vh$ (2) |
| Calculate velocity and position of nodes in each hidden layer | $vn = vn + c1 * \text{rand} * (n\text{Best} - n) + c2 * \text{rand} * (gn\text{Best} - n)$(3)<br>$n = n + vn$ (4) |
| Calculate velocity and position of number of weights in memory cell and in each gate | $vw = vw + c1 * \text{rand} * (w\text{Best} - w) + c2 * \text{rand} * (gw\text{Best} - w)$(5)<br>$w = w + vw$ (6) |
| Calculate velocity and position of number of bias in memory cell and in each gate | $vb = vb + c1 * \text{rand} * (b\text{Best} - b) + c2 * \text{rand} * (gb\text{Best} - b)$(7)<br>$b = b + vb$ (8) |
| Calculate velocity and position of activation function | $vaf = vaf + c1 * \text{rand} * (af\text{Best} - af) + c2 * \text{rand} * (gaf\text{Best} - af)$(9)<br>$af = af + vaf$ (10) |

***Algorithm #1: Pre-Processing***

*Input: Air pollution dataset include multi-station and six concentration of (PM2.5, PM10, NO2, CO, O3, and SO2)*

*Output: Split station [id_station] have clean data in range [0, 1]*

```
// Split air pollution dataset according [id_station]
    For all samples in air pollution dataset
        IF station_name = id_station
            create a file called station_name
            Put all the concentrations related to station_name in that file
        End IF
    End for
// Treatment missing values
    For each row in air pollution dataset
        For each column in air pollution dataset
            IF value of column = null
                dropping  the row contains that columns
            End IF
        End for
    End for
// Apply normalization
    For each row in air pollution dataset
        For each column in air pollution dataset
            Compute MinMaxScaler    // MinMaxScaler= (x − min x) / (max x − min x)
        End for
    End for
End pre-processing
```

**Algorithm#2: DSN-PS**

***Input:*** *Training part of dataset based on [station-id ] after complete preprocessing*

***Output:*** *Best Structure and parameters of DLSTM*

***Initialization****: pi: Position of population, problem size, Population Size,*
*f: Objective function, $v_i$: Velocity of population,*
*f(phBest), f(pnBest), f(pwBest), f(pbBest), f(pafBest)=0*

```
For i in rang(1 to Max_interation)
    For node of hidden layer h in H do
        Calculate fitness function as equation (1)
        IF fph is better than f(phBest)
            phBest = ph
        End IF
    End for
    For the number of the nodes of each hidden layer n in N do
        Calculate fitness function as equation (2)
        IF fpn is better than f(pnBest)
            pnBest = pn
        End IF
    End for
    For random weights w in W do
        Calculate fitness function
        IF fpw is better than f(pwBest)
            pwBest =pw
        End IF
    End for
    For random bias b in B do
        Calculate fitness function
        IF fb is better than f(bBest)
            pbBest = pb
        End IF
    End for
    Call activation function af in AF
        Calculate fitness function
        IF fpaf is better than f(pafBest)
            pafBest = paf
        End IF
    ghBest = best h in H
    For the number of hidden layer h in H do
        Calculate hidden layers velocity as equation (3)
        Calculate hidden layers position as equation (4)
    End for
    gnBest = best n in N
    For the number of nodes in each hidden layer n in N do
        Calculate the velocity of nodes in each hidden layer as equation (5)
        Calculate the position of nodes in each hidden layer as equation (6)
    End for
    gwBest = best w in W
    For random weights w in W do
        Calculate the velocity of weight i in the memory cell and each gate as equation (7)
        Calculate the position of weight i in the memory cell and each gate as equation (8)
    End for
    gbBest = best b in B
    For random bias b in B do
        Calculate the velocity of bias i in the memory cell and each gate as equation (9)
        Calculate the position of bias i in the memory cell and each gate as equation (10)
    End for
    gafBest = best af in AF
    Calculate the velocity of the activation function as equation (11)
    Calculate the position of the activation function as equation (12)
End for
End DNS-PS
```

**Algorithm #3: Essential Components of DLSTM**

***Input:*** *X: Dataset of air pollution & DSN-PS*

***Output****: Prediction values of ($PM_{2.5}$, $PM_{10}$, $O_3$, $NO_2$, CO, and SO2)*

***// The forward components***

***For each time (t) in a dataset of air pollution***

***//Compute: $a_t$, $i_t$, $f_t$, $o_t$, $STATE_t$, $OUT_t$***

$a_t = tanh(Wa.\ X_t + U_a.\ out_{t-1} + b_a)$ // Memory Cell

$i_t = \sigma(Wi.\ X_t + U_i.\ out_{t-1} + b_i)$ // Input gate

$f_t = \sigma\ (Wf.\ X_t + U_f.\ out_{t-1} + b_f)$ // forgate gate

$O_t = \sigma\ (Wo.\ X_t + U_o.\ out_{t-1} + b_o)$ // Output gate

$state_t = a_t \odot i_t + f_t\ O\ state_{t-1}$ // internal state

$Out_t = tanh(state_t)\ O\ o_t$ //output cell

***End*** *for*

***//The backward components.***

***For each time (t) update***

$\delta out_t = \Delta_t + \Delta out_t$

$state_t = \delta out_t \odot o_t \odot \left(1 - tanh^2(state_t)\right) + \delta state_{t+1} \odot f_{t+1}$

***//Update of Memory Cell Input gate, Forget gate and Output gate.***

$\delta a_t = \delta state_t \odot i_t \odot (1 - a_t^2)$ //update memory cell

$\delta i_t = \delta state_t \odot a_t \odot i_t \odot (1 - i_t)$ // update input gate

$\delta f_t = \delta state_t \odot state_{t-1} \odot f_t \odot (1 - f_t)$ //update Forget gate

$\delta o_t = \delta out_t \odot \tanh(state_t) \odot o_t \odot (1 - o_t)$ //update *Output gate*

$\delta x_t = W^t . \delta state_t$ //update input

$\delta out_{t-1} = U^t . \delta state_t$ // update output

***END*** *for*

***//The final updates to the internal parameters is compute***

$\delta W = \sum_{t=0}^{T} \delta gates_t \otimes x_t$ *//update the weight of input*

$\delta U = \sum_{t=0}^{T} \delta gates_{t+1} \otimes out_t$ *//update the weight of* recurrent connections

$\delta b = \sum_{t=0}^{T} \delta gates_{t+1}$ *//update bias*

***End Essential Components of LSTM***

## 3.2 Running the IFCsAP model

We will train and predict concentrations movements for several epochs and see whether the predictions get better or worse over time. The algorithm is shown how execution the IFCsAP model.

**Algorthim#4: Execution of IFCsAP MODEL**

***Input:*** *Datasets split into part training and testing*

***Output:*** *Average values of AMAPE to 25 days*

***For*** *each iteration*

  ***For*** *all samples in the training dataset*

    *Unroll a set of num_unrollings batches*

    *Train the DLSTM with the unrolled batches*

    *Compute the average training loss*

  ***End for***

  ***For*** *each record in the testing dataset*

    *Update the IFCSAP MODEL state*

    *Found num_unrollings data points*

    *Make forecasting for forecaster _once steps continuously,*

    *Current input is previous prediction*

  ***End for***

***End for***

***End Execution of IFCsAP MODEL***

### 3.3 Evaluation stage

The symmetric mean absolute percentage error (SMAPE) is used in this paper as measured to determine the accuracy and robust of the predictor.

SAMPE =

N: number of samples. : forecast value.: Actual value. t: every fitted point.

Set SMAPE score as 0. If both values predict and actual are 0 actual value and forecast value are both 0. In each station forecasting the concentration levels "PM2.5, PM10, NO2, CO, O3 and SO2" to the next 48 h. We can calculate the values of this measure daily contagious through one moth then sort these values and compute the average of 25 lowest daily SMAPE scores. The main steps of evaluation shown with details in algorithm 5.

**Algorthim#5: Evaluation**
***Input:*** *Testing dataset*
***Output:*** *Average error rate based on SMAPE for 25 days*

```
For number (id_station)
    For 1 to 30                          // 30 represent the total number of days
        For 1 to 24                      // 24 represent the total number of hours
            SMAPE (actual, predicted):
            dividend= np.abs (np.array (actual) - np.array (predicted))
            denominator = np.array (actual) + np.array (predicted)
            np.mean (np.divide (dividend, denominator)
        End for
    End for
    S = the summation of SMAPE in less 25 days
    Average of SMAPE = S /25
End for
End Evaluation
```

## 4 Results of IFCsAP model

The results and justification of it will explain with details in this section.

### 4.1 Pre-processing

This stage consists of multi-steps performance on the dataset after collecting it, each step handles the dataset from one problem as we will be discussed later.

#### 4.1.1 Split station

The second column of Table 4 shows the result of splitting the dataset based on the name of station, where each station saves in a separated file hold the name of it.

#### 4.1.2 Missing values [8]

Missing values one of the problem effect in the final results of any model. Spatially the prediction model, where all researches know the result of predictor become more accuracy of that predictor build based on true values otherwise the results become not truest. Therefore, in that model, we will drop any record have missing values in each station. In general, the station has different rate of missing value as explained in Table 2 column three and Fig. 5.

Table 2 explains the dataset after split it into 35 stations have the same number of record 8886 and six features.

**Table 2** Ratio after pre-processing the missing values

| No | Name of station | # Record after drop | Ratio |
|---|---|---|---|
| 1 | Daxing_Aq | 6806 | 23% |
| 2 | Mentougou_Aq | 6639 | 25% |
| 3 | Badaling_Aq | 6566 | 26% |
| 4 | Fangshan_Aq | 6536 | 26% |
| 5 | Dongsi_Aq | 6520 | 27% |
| 6 | Tiantan_Aq | 6470 | 27% |
| 7 | Fengtaihuayuan_Aq | 6385 | 28% |
| 8 | Gucheng_Aq | 6419 | 28% |
| 9 | Pingchang_Aq | 6381 | 28% |
| 10 | Yungang_Aq | 6385 | 28% |
| 11 | Guanyuan_Aq | 6284 | 29% |
| 12 | Wanliu_Aq | 6284 | 29% |
| 13 | Aotizhongxin_Aq | 6134 | 31% |
| 14 | Yizhuang_Aq | 6092 | 31% |
| 15 | Dingling_Aq | 6003 | 32% |
| 16 | Miyun_Aq | 6005 | 32% |
| 17 | Wanshouxigong_Aq | 6068 | 32% |
| 18 | Nongzhanguan_Aq | 5977 | 33% |
| 19 | Yanqin_Aq | 5994 | 33% |
| 20 | Beibuxinqu_Aq | 5842 | 34% |
| 21 | Shunyi_Aq | 5893 | 34% |
| 22 | Huairou_Aq | 5815 | 35% |
| 23 | Tongzhou_Aq | 5660 | 36% |
| 24 | Xizhimenbei_Aq | 5554 | 37% |
| 25 | Qianmen_Aq | 5389 | 39% |
| 26 | Yongdingmennei_Aq | 5275 | 41% |
| 27 | Yongledian_Aq | 4997 | 44% |
| 28 | Pinggu_Aq | 4905 | 45% |
| 29 | Yufa_Aq | 4832 | 46% |
| 30 | Nansanhuan_Aq | 4726 | 47% |
| 31 | Dongsihuan_Aq | 4638 | 48% |
| 32 | Miyunshuiku_Aq | 4604 | 48% |
| 33 | Liulihe_Aq | 4411 | 50% |
| 34 | Donggaocun_Aq | 3956 | 55% |
| 35 | Zhiwuyuan_Aq | 3945 | 56% |

Also show the dataset after handle the missing values with the rate of dropping. Also, Fig. 6 shows the percentage of records have missing values in each station.

#### 4.1.3 Normalization

Normalization dataset based on MinMaxScaler scales to become in the range [0 and 1] [20, 25]. This is a necessary step for the proposed predictor. The main purpose of the normalization stage is to make all the values in the same range with save the natural of each feature in that dataset.

#### 4.1.4 Split the dataset

Cross-validation is the best techniques for evaluating the performance of a given model. Because badly selected samples for training and testing affects the performance badly, cross-validation has different methods for wisely selecting the best samples for training and testing a given model. As shown in Table 3 as attached in "Appendix" and Fig. 7.

Table 4 shows idea of cross-validation which, in this paper, used ten cross-validation for each station to determine the best number of samples will used from training dataset to build model and from testing dataset to evaluation of the model.

We note that the station with the highest percentage of missing values has a very high SMAPE score compared to stations with the lowest percentage of missing values. We conclude that using the drooping process will make the predictor results more accurate compared to other methods used to process missing values (Fig. 8).

### 4.2 DSN-PS

Select the suitable parameters of any deep learning algorithm is consider one of the main challenges in the science, in general, all known LSTM take a very long time in implementation to give the result; therefore, this section shows how DSN-PS solve this problem and exceed this challenge. The optimal structure with main parameters was find to DLSTM.

In other words, values determined as hidden layers number, nodes in each hidden layer, weights among layers, the bias, and activation function type of the deep learning network are essential parameters that fundamentally affect DLSTM performance. In general, all the network based on the try and error principle in select the parameters of it, while this led to long time on implementation that network. Therefore, the main parameters of DLSTM result from DSN-PS as shown in Table 5. While Table 6 shows the best parameters that represent the structure of DLSTM and compare with the parameters of traditional LSTM.

Table 6 shows best parameters (# hidden layers, #nodes in each hidden layer, weights, bias, and activation function) resulted from the DSN-PS algorithm that represents the initial structure of the DLSTM (Table 7).

### 4.3 DLSTM

DLSTM is mainly based on the LSTM algorithm, which is capable of handling large data and retains data for long periods because each cell contains memory. In this stage, forward the parameters result from DNS-PS to DLSTM that represent the structure of it with the dataset of that

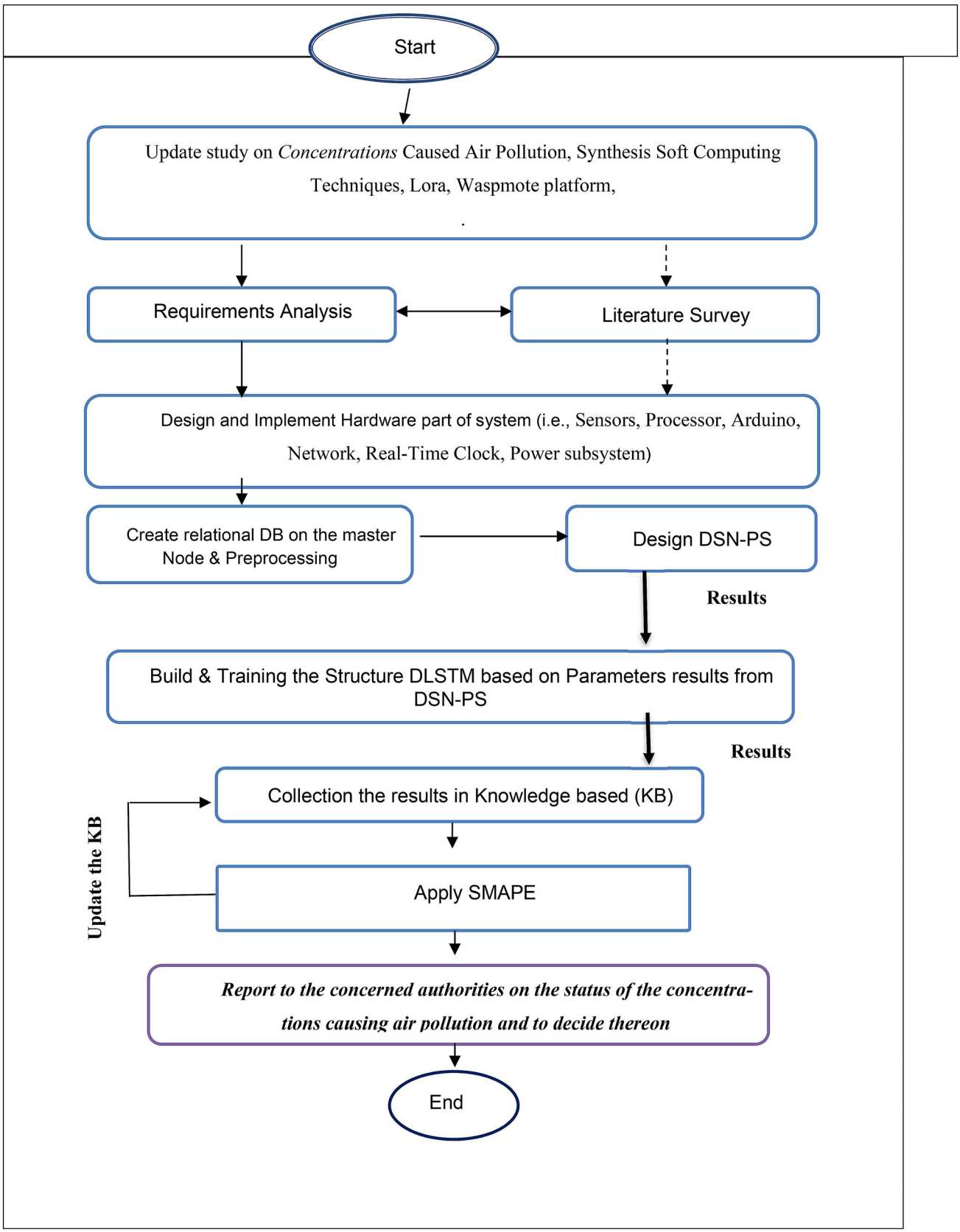


**Fig. 5** Flowchart of research work activities

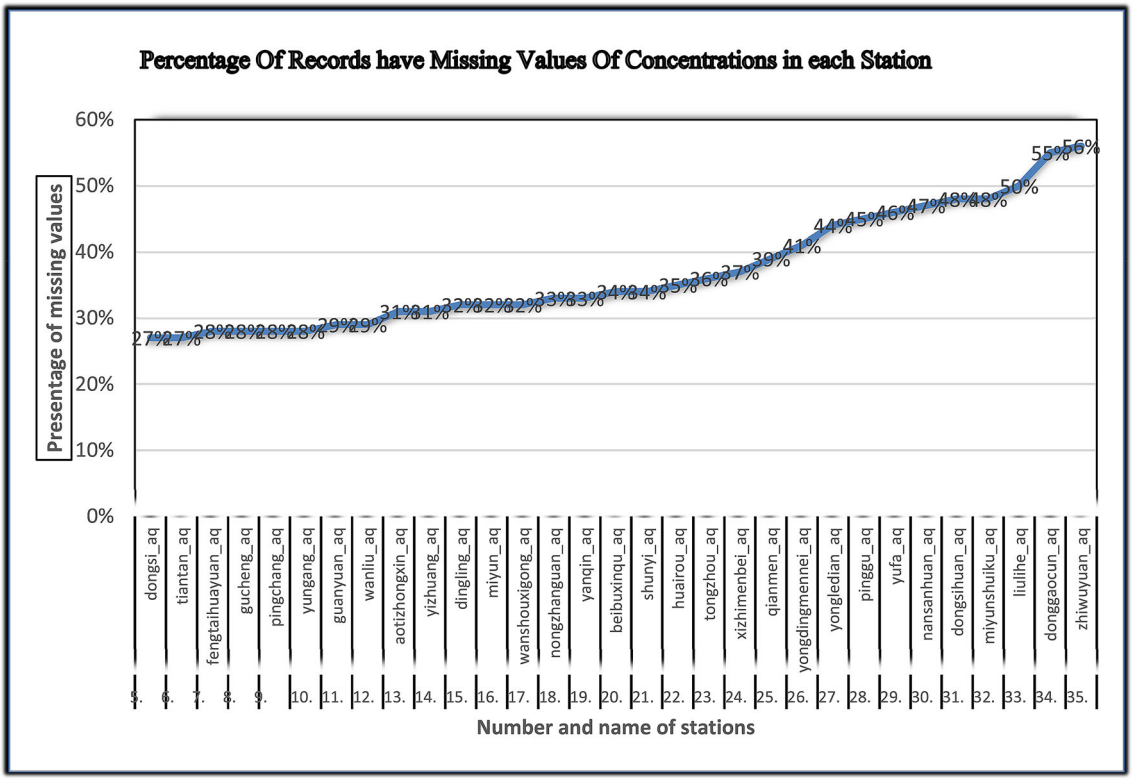


**Fig. 6** Percentage of records have missing values in each station

station generated from the best split of ten cross-validations to represent training of DLSTM. Compute the prediction values for each station (Station #1… Station #35) based on the best split result from ten cross-validations. The best parameters result from DNS-PSO as structure of DLSTM represent one input layer have six nodes each node represent one of six constructions; one hidden layer contain 250 nodes, one output layer. All other parameters and activation function described in Table 8. Also, we used 150 iteration in each iteration we enter batch size 24.

Compare the actual and prediction values results from DLSTM for first station shown in Fig. 9.

Compare the prediction values Station #18 based on the best split result from ten cross-validation and compare with the real values shown in Fig. 10.

Compare the prediction values Station #34 based on the best split result from ten cross-validation and compare with the real values shown in Fig. 11.

### 4.4 SMAPE evaluation

After, build the DLSTM based on the training dataset for each station, the model evaluated through compute SMAPE for testing dataset.

The result score of each concentration is the average of 25 lowest daily SMAPE scores. If a concentration misses a day, the score of this concentration on that day will be imputed by the baseline score. As shown in "Appendix" under Table 8.

## 5 Compare between traditional LSTM and IFCsAP based on the values of SMAPE

To explain the successful of IFCsAP model, we compare the result values of SMAPE come from the traditional LSTM and IFCsAP. As shown in Table 9.

The above table showed the result of SMAPE of IFCsAP Model, in comparison with the result of SMAPE of traditional LSTM. Which used the same dataset from the

**Table 3** Apply cross-validation for each station

| Name of station | Training–testing | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sample #1 90%-10% | Sample #2 80%-20% | Sample #3 70%-30% | Sample #4 60%-40% | Sample #5 50%-50% | Sample #6 40%-60% | Sample #7 30%-70% | Sample #8 20%-80% | Sample #9 10%-90% |
| Aotizhongxin_Aq | 5520–613 | 4907–1226 | 4293–1840 | 3680–2453 | 3066–3067 | 2453–3680 | 1840–4293 | 1226–4907 | 613–5520 |
| | SMAPE | | | | | | | | |
| | 0.34833 | **0.32599** | 0.35920 | 0.34913 | 0.36971 | 0.35780 | 0.33877 | 0.32898 | 0.31553 |
| Badaling_Aq | 5909–656 | 5252–1313 | 4596–1969 | 3939–2626 | 3283–3282 | 2626–3939 | 1969–4596 | 1313–5252 | 656- 5909 |
| | SMAPE | | | | | | | | |
| | **0.22043** | 0.24514 | 0.27469 | 0.32280 | 0.30192 | 0.30951 | 0.28630 | 0.30145 | 0.29601 |
| Beibuxinqu_Aq | 5256–585 | 4673–1168 | 4088–1753 | 3504–2337 | 2920–2921 | 2336–3505 | 1752–4089 | 1168–4673 | 584–5257 |
| | SMAPE | | | | | | | | |
| | **0.37632** | 0.37849 | 0.42684 | 0.42765 | 0.39135 | 0.39482 | 0.38102 | 0.39621 | 0.33604 |
| Daxing_Aq | 6125–680 | 5444–361 | 4764–2041 | 4083–2722 | 3403–3402 | 2722–4083 | 2041–4764 | 1361–5444 | 680- 6125 |
| | SMAPE | | | | | | | | |
| | 0.38115 | **0.34451** | 0.37958 | 0.37754 | 0.36309 | 0.38190 | 0.35811 | 0.36822 | 0.34966 |
| Dingling_Aq | 5402–600 | 4802–1200 | 4202–1800 | 3601–2401 | 3001–3001 | 2401–3601 | 1800–4202 | 1200–4802 | 600- 5402 |
| | SMAPE | | | | | | | | |
| | **0.48551** | 0.52657 | 0.56950 | 0.56276 | 0.56138 | 0.57259 | 0.51382 | 0.47585 | 0.48068 |
| Donggaocun_Aq | 3560–395 | 3164–791 | 2769–1186 | 2373–1582 | 1978–1977 | 1582–2373 | 1186–2769 | 791–3164 | 395–3560 |
| | SMAPE | | | | | | | | |
| | 0.56709 | 0.47908 | **0.47250** | 0.49197 | 0.51633 | 0.49115 | 0.49124 | 0.51901 | 0.49028 |
| Dongsi_Aq | 5868–651 | 5216–1303 | 4564–1955 | 3912–2607 | 3260–3259 | 2608–3911 | 1956–4563 | 1304–5215 | 652–5867 |
| | SMAPE | | | | | | | | |
| | 0.37109 | 0.32762 | 0.34534 | 0.33234 | **0.31108** | 0.39653 | 0.73601 | 0.38420 | 0.39063 |
| Dongsihuan_Aq | 4174–463 | 3710–927 | 2782–1855 | 2782–1855 | 2319–2318 | 1855–2782 | 1955–4564 | 927–3710 | 463–4174 |
| | SMAPE | | | | | | | | |
| | 0.30586 | **0.25909** | 0.34534 | 0.32434 | 0.31514 | 0.36185 | 0.37994 | 0.37932 | 0.32896 |
| Fangshan_Aq | 5881–654 | 5228–1307 | 4574–1961 | 3921–2014 | 3267–3268 | 2614–3921 | 1960–4575 | 1307–5228 | 653–5882 |
| | SMAPE | | | | | | | | |
| | **0.34521** | 0.37708 | 0.40295 | 0.39091 | 0.42268 | 0.35006 | 0.42037 | 0.34782 | 0.35412 |
| Fengtaihuayuan_Aq | 5745–639 | 5107–1277 | 4468–1916 | 3830–2554 | 3192–3192 | 2553–3831 | 1915–4469 | 1276–5108 | 638- 5746 |
| | SMAPE | | | | | | | | |
| | **0.41832** | 0.44171 | 0.44235 | 0.55864 | 0.44715 | 0.47422 | 0.41944 | 0.42332 | 0.42002 |
| Guanyuan_Aq | 5654–629 | 5026–1257 | 4398–1885 | 3769–2514 | 3141–3142 | 2513–3770 | 1884–4399 | 1256–5027 | 628–5655 |
| | SMAPE | | | | | | | | |
| | 0.68366 | 0.45748 | 0.54917 | 0.47553 | **0.43397** | 0.44676 | 0.48992 | 0.45837 | 0.48204 |
| Gucheng_Aq | 5776–642 | 5134–1284 | 4492–1926 | 3850–2568 | 3209–3209 | 2567–3851 | 1925–4493 | 1283–5135 | 641–5777 |
| | SMAPE | | | | | | | | |
| | 0.29839 | 0.31746 | **0.29639** | 0.29969 | 0.32755 | 0.30655 | 0.29722 | 0.29715 | 0.33808 |
| Huairou_Aq | 5232–582 | 4651–1163 | 4069–1745 | 3488–2326 | 2907–2907 | 2325–3489 | 1744–4070 | 1162–4652 | 581- 5233 |
| | SMAPE | | | | | | | | |
| | 0.50856 | 0.47332 | 0.46545 | 0.46634 | **0.45716** | 0.47513 | 0.49482 | 0.48409 | 0.46173 |
| Liulihe_Aq | 3969–441 | 3528–882 | 3087–1323 | 2646–1764 | 2205–2205 | 1764–2646 | 1323–3087 | 882- 3528 | 441- 3969 |
| | SMAPE | | | | | | | | |
| | **0.29100** | 0.30342 | 0.34089 | 0.35732 | 0.35035 | 0.33702 | 0.33319 | 0.34248 | 0.34783 |
| Mentougou_Aq | 5974–664 | 5310–1328 | 4646–1992 | 3982–2656 | 3319–3319 | 2655–3983 | 1991–4647 | 1327–5311 | 663- 5975 |
| | SMAPE | | | | | | | | |
| | 0.33704 | **0.31976** | 0.34966 | 0.38730 | 0.32720 | 0.34968 | 0.32988 | 0.32414 | 0.32786 |
| Miyun_Aq | 5403–601 | 4803–1201 | 4202–1802 | 3602–2402 | 3002–3002 | 2401–3603 | 1801–4203 | 1200–4804 | 600- 5404 |

**Table 3** (continued)

| Name of station | Training–testing | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sample #1 90%-10% | Sample #2 80%-20% | Sample #3 70%-30% | Sample #4 60%-40% | Sample #5 50%-50% | Sample #6 40%-60% | Sample #7 30%-70% | Sample #8 20%-80% | Sample #9 10%-90% |
| | SMAPE | | | | | | | | |
| | 0.43071 | 0.42796 | 0.44320 | **0.41744** | 0.41894 | 0.43518 | 0.47480 | 0.49351 | 0.42136 |
| Miyunshuiku_Aq | 4142–461 | 3682–921 | 3222–1381 | 2761–1842 | 2301–2302 | 1841–2762 | 1380–3223 | 920- 3683 | 460- 4143 |
| | SMAPE | | | | | | | | |
| | 0.58134 | **0.55469** | 0.63117 | 0.59486 | 0.56510 | 0.55728 | 0.55989 | 0.56633 | 0.57948 |
| Nansanhuan_Aq | 4252–473 | 3780–945 | 3307–1418 | 2835–1890 | 2362–2363 | 1890–2835 | 1417–3308 | 945- 3780 | 472- 4253 |
| | SMAPE | | | | | | | | |
| | 0.34222 | 0.32792 | 0.37368 | **0.22149** | 0.45042 | 0.40474 | 0.35022 | 0.31506 | 0.33891 |
| Nongzhanguan_Aq | 5378–598 | 4780–1196 | 4183–1793 | 3585–2391 | 2988–2988 | 2390–3586 | 1792–4184 | 1195–4781 | 597- 5379 |
| | SMAPE | | | | | | | | |
| | 0.35150 | **0.32463** | 0.38396 | 1.2097 | 0.34598 | 0.34410 | 0.33725 | 0.32831 | 0.34449 |
| Pingchang_Aq | 5742–638 | 5104–1276 | 4466–1914 | 3828–2552 | 3190–3190 | 2552–3828 | 1914–4466 | 1276–5104 | 638–5742 |
| | SMAPE | | | | | | | | |
| | **0.38221** | 0.39623 | 0.41118 | 0.42677 | 0.39948 | 0.43234 | 0.39079 | 0.41750 | 0.41457 |
| Pinggu_Aq | 4413–491 | 3923–981 | 3432–1472 | 2942–1962 | 2452–2452 | 1961–2943 | 1471–3433 | 980–3924 | 4414–490 |
| | SMAPE | | | | | | | | |
| | 0.42887 | **0.35100** | 0.45153 | 0.41323 | 0.38462 | 0.38099 | 0.36968 | 0.35607 | 0.35181 |
| Qianmen_Aq | 4849–539 | 4310–1078 | 3771–1617 | 3232–2156 | 2694–2694 | 2155–3233 | 1616–3772 | 1077- 311 | 538- 4850 |
| | SMAPE | | | | | | | | |
| | 0.53473 | 0.48188 | 0.45428 | 0.89437 | **0.39968** | 0.40737 | 0.50851 | 0.40141 | 0.40571 |
| Shunyi_Aq | 5302–590 | 4713–1179 | 4124–1768 | 3535–2357 | 2946–2946 | 2356–3536 | 1767–4125 | 1178–4714 | 589- 5303 |
| | SMAPE | | | | | | | | |
| | 0.40456 | 0.42718 | 0.39915 | 0.45804 | **0.39386** | 0.39555 | 0.47355 | 0.39463 | 0.39796 |
| Tiantan_Aq | 5822–647 | 5175–1294 | 4528–1941 | 3881–2588 | 3234–3235 | 2587–3882 | 1940–4529 | 1293–5176 | 646–5823 |
| | SMAPE | | | | | | | | |
| | 0.53815 | 0.48985 | 0.46503 | 0.43918 | **0.43291** | 0.43770 | 0.43587 | 0.48286 | 0.49288 |
| Tongzhou_Aq | 5093–566 | 4527–1132 | 3961–1698 | 3395–2264 | 2829–2830 | 2263–3396 | 1697–3962 | 1131–4528 | 565–5094 |
| | SMAPE | | | | | | | | |
| | 0.32452 | **0.30688** | 0.75632 | 0.34207 | 0.33272 | 0.37680 | 0.31931 | 0.31494 | 0.31636 |
| Wanliu_Aq | 5654–629 | 5026–1257 | 4398–1885 | 3769–2514 | 3141–3142 | 2513–3770 | 1884–3499 | 1256–5027 | 5655–628 |
| | SMAPE | | | | | | | | |
| | 0.45249 | 0.41911 | 0.45745 | **0.38290** | 0.43135 | 0.41561 | 0.38329 | 0.38339 | 0.38896 |
| Wanshouxigong_Aq | 5460–607 | 4853–1214 | 4246–1821 | 3640–2427 | 3033–3034 | 2426–3641 | 1820–4247 | 1213–4854 | 606- 5461 |
| | SMAPE | | | | | | | | |
| | 0.43756 | **0.37226** | 0.37845 | 0.39268 | 0.38290 | 0.37232 | 0.37923 | 0.37357 | 0.82641 |
| Xizhimenbei_Aq | 4997–556 | 4442–1111 | 3887–1666 | 3331–2222 | 2776–2777 | 2221–3332 | 1665–3888 | 1110–4443 | 555- 4998 |
| | SMAPE | | | | | | | | |
| | **0.27052** | 0.27459 | 0.28166 | 0.31190 | 0.27692 | 0.27862 | 0.27753 | 0.27397 | 0.27912 |
| Yanqin_Aq | 5393–600 | 4794–1199 | 4195–1798 | 3595–2398 | 2996–2997 | 2397–3596 | 1797–4196 | 1198–4795 | 599–5394 |
| | SMAPE | | | | | | | | |
| | 0.33129 | **0.31195** | 0.33281 | 0.38399 | 0.36814 | 0.35996 | 0.37810 | 0.37346 | 0.37667 |
| Yizhuang_Aq | 5481–610 | 4872–1219 | 4263–1828 | 3654–2437 | 3045–3046 | 2436–3655 | 1827–4264 | 1218–4873 | 609- 5482 |
| | SMAPE | | | | | | | | |
| | 0.59806 | 0.51447 | 0.64442 | **0.19580** | 0.55716 | 0.36780 | 0.30094 | 0.44786 | 0.45168 |
| Yongdingmennei_Aq | 4746–528 | 4219–1055 | 3691–583 | 3164–2110 | 2637–2637 | 2109–3165 | 1582–3692 | 1054–4220 | 527–4747 |
| | SMAPE | | | | | | | | |

**Table 3** (continued)

| Name of station | Training–testing | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Sample #1 90%-10% | Sample #2 80%-20% | Sample #3 70%-30% | Sample #4 60%-40% | Sample #5 50%-50% | Sample #6 40%-60% | Sample #7 30%-70% | Sample #8 20%-80% | Sample #9 10%-90% |
| | 0.34736 | 0.34834 | 0.34725 | 0.33133 | **0.32434** | 0.33073 | 0.33132 | 0.32926 | 0.33506 |
| Yongledian_Aq | 4496–500 | 3996–1000 | 3497–1499 | 2997–1999 | 2498–2498 | 1998–2998 | 1498–3498 | 999–3997 | 499–4497 |
| | SMAPE | | | | | | | | |
| | **0.43421** | 0.43423 | 0.44739 | 0.45287 | 0.45177 | 0.44419 | 0.44299 | 0.44033 | 0.44038 |
| Yufa_Aq | 4347–484 | 3864–967 | 3381–1450 | 2898–1933 | 2415–2416 | 1932–2899 | 1449–3382 | 966–3865 | 483–4348 |
| | SMAPE | | | | | | | | |
| | 0.40199 | **0.36498** | 0.36876 | 0.39849 | 0.39930 | 0.39239 | 0.45638 | 0.41923 | 0.39684 |
| Yungang_Aq | 5745–639 | 5107–1277 | 4468–1916 | 3830–2554 | 3193–3192 | 2553–3831 | 1915–4469 | 1276–5108 | 638–5746 |
| | SMAPE | | | | | | | | |
| | 0.28616 | 0.28983 | **0.26099** | 0.28553 | 0.29743 | 0.29826 | 0.29826 | 0.28685 | 0.28849 |
| Zhiwuyuan_Aq | 3549–395 | 3155–789 | 2760–1184 | 2366–1578 | 1972–1972 | 1577–2367 | 1183–2761 | 788–3156 | 394–3550 |
| | SMAPE | | | | | | | | |
| | 0.57043 | 0.66970 | 0.60734 | 0.61349 | **0.54953** | 0.55605 | 0.55614 | 0.54986 | 0.55225 |

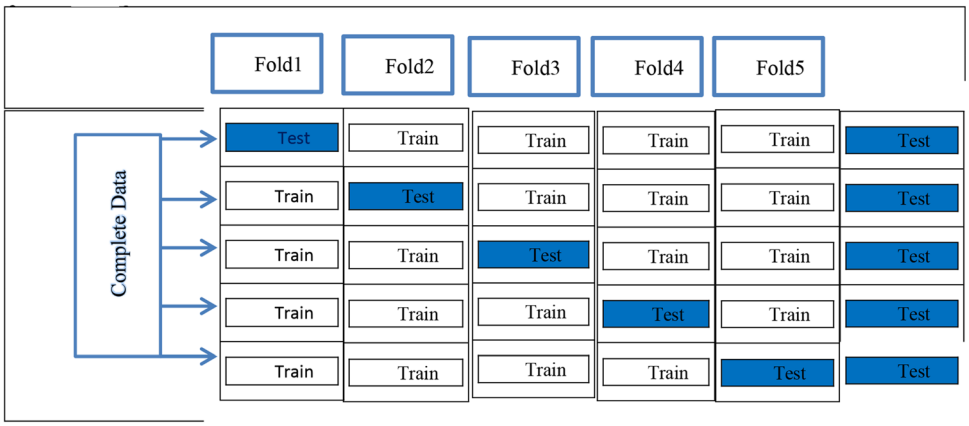


**Fig. 7** Distribution of dataset based on five cross-validation

pre-processing stage (i.e., the dropping, normalization and the same split of training and testing resulting from ten cross-validation) were applied at each station. We found that the results SMAPE of IFCsAP model are better than traditional LSTM as shown in Fig. 12.

## 6 Summary

Air quality index dataset is a huge data needed to intelligent and deep computation to extract a useful pattern from it. The advantage of this data set is diverse and large in size, resulting in accurate and reliable decisions. In addition, the data used in this thesis were obtained from more than one station and this in itself is considered a challenge

**Table 4** The best split for the dataset of each station

| No. station | Best split based on training and testing | Number of samples | SMAPE | No. station | Best split based on training and testing | Number of samples | SMAPE |
|---|---|---|---|---|---|---|---|
| 1 | 80%–20% | 4907–1226 | 0.32599 | 19 | 80%–20% | 4780–1196 | 0.32463 |
| 2 | 90%–10% | 5909–656 | 0.22043 | 20 | 90%–10% | 5742–638 | 0.38221 |
| 3 | 90%–10% | 5256–585 | 0.37632 | 21 | 80%–20% | 3923–981 | 0.351 |
| 4 | 80%–20% | 5444–361 | 0.33451 | 22 | 50%–50% | 2694–2694 | 0.39968 |
| 5 | 90%–10% | 5402–600 | 0.48551 | 23 | 50%–50% | 2946–2946 | 0.39968 |
| 6 | 30%–70% | 2769–1186 | 0.4725 | 24 | 50%–50% | 3234–3235 | 0.43291 |
| 7 | 50%–50% | 3260–3259 | 0.31108 | 25 | 80%–20% | 4527–1132 | 0.30688 |
| 8 | 80%–20% | 3710–927 | 0.25909 | 26 | 60%–40% | 3769–2514 | 0.3829 |
| 9 | 90%–10% | 5881–654 | 0.34521 | 27 | 80%–20% | 4853–1214 | 0.37726 |
| 10 | 90%–10% | 5745–639 | 0.41832 | 28 | 90%–10% | 4997–556 | 0.27052 |
| 11 | 50%–50% | 3141–3142 | 0.43397 | 29 | 80%–20% | 4794–1199 | 0.31195 |
| 12 | 30%–70% | 4492–1926 | 0.29639 | 30 | 60%–40% | 3654–2437 | 0.1958 |
| 13 | 50%–50% | 2907–2907 | 0.45716 | 31 | 50%–50% | 2637–2637 | 0.32434 |
| 14 | 90%–10% | 3969–441 | 0.291 | 32 | 90%–10% | 4496–500 | 0.43421 |
| 15 | 80%–20% | 5310–1328 | 0.31976 | 33 | 80%–20% | 3864–967 | 0.36498 |
| 16 | 60%–40% | 3602–2402 | 0.41744 | 34 | 30%–70% | 4468–1916 | 0.26099 |
| 17 | 80%–20% | 3682–921 | 0.55469 | 35 | 50%–50% | 1972–1972 | 0.54953 |
| 18 | 60%–40% | 2835–1890 | 0.22149 | | | | |

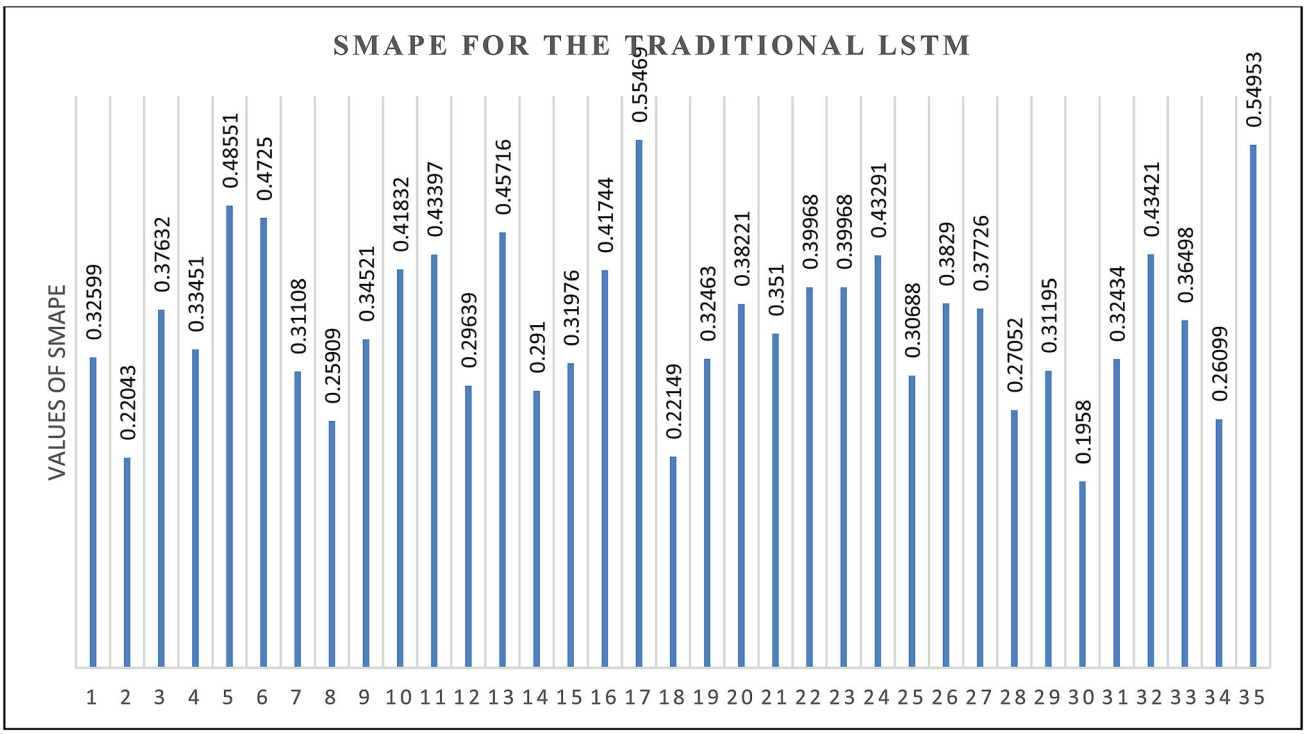


**Fig. 8** SMAPE based on the traditional LSTM

**Table 5** The parameters of DSN-PS

| Parameter | Value |
|---|---|
| Learning coefficients: c1, c2 | C1 = 0.87 while C2 = 2.53 |
| Maximum number of iterations | 30 |
| *r*, *r* | R1 = 0.6, While, R2 = 0.4 |

in building a stable prediction system for behaviors. Limitation of this dataset contains on concentrations that cause air pollution are usually unequal and unknown to non-experts, which contain missing value and taken from different stations in terms of the environment assigned to those stations.

DSN-PS is determined the parameters and activation function of DLSTM, the advantage of DSN-PS is the time of execution LSTM will be reduced, limitation of DSN-PS will increase complex of LSTM.

DLSTM is a develop of LSTM by DSN-PS, POS used to determine the optimal (number of hidden layers, number of nodes in each hidden layer, weight, bias, and activation function), the advantage of DLSTM capable to deal with huge data and contain memory cell to save information at the long term, the limitation of DSTM contain on huge number of parameters.

Evaluation is the process of calculating the amount of error from the actual value and its predicted value, there are different types of error measures: including prediction (i.e., MSE, RMSE, MAE, MAPE and, etc.) and coefficient matrix (i.e., accuracy, F, FP, etc.). While in this research, use SMAPE Evaluation.

- How particle swarm can be useful in building a recurrent neural network (RNN)?

  PSO works to modify the behavior of each in a particular environment gradually, depending on the behavior of their neighbors until they are obtained the optimal solution.

  On the other hand, the neural networks use the principle of the try and error in the selection of the basic parameters of their own and modified gradually to reach the values accepted for those parameters.

  Depending on the PSO and neural networks of the above subject, we used the PSO principle to find the optimal parameters and the activation function of the neural network.
- How to build a multi-layer model with a combination of two technologies LSTM-RNN with particle swarm?

  Through, building new predictor called IFCsAP that combining between the DSN-PS and the DLSTM. Where DSN-PS used to find the best structure with parameter to LSTM while DLSTM used to predict the rate Concentrations of air pollution.
- IS SMAPE measure enough to evaluate the results of suggesting predictor?

  Yes, The SMAPE is sufficient to evaluate the results of the predictor within the next 48 h.
- What is the benefit result from building predictor by combination between DSN-PS and DLSTM?

By combining DNS-PS and DLSTM, reduce the execution time by defining network parameters but at the same time will increase the computational complexity.

**Table 6** The best parameters of DLSTM results from apply DSN-PS compare with the parameters of traditional LSTM

| Parameter | DLSTM | Traditional LSTM |
|---|---|---|
| # Hidden layers | 1 | 1 |
| # Nodes in each layer | 250 | 300 |
| Weights in the memory cell and each gate | [[−0.08035341, 0.0914277, −0.06338812, …, 0.02485519, 0.00429418, 0.0344343]. [−0.05277034, 0.02425835, −0.08560476, …, −0.02298651, −0.09942206, −0.0467921], [0.05811697, −0.13458245, 0.04627861, …, 0.05916523,0.00762758, −0.05373584], [−0.07273816, 0.01714688, −0.07424279, …, 0.02360445, −0.0214516, −0.04725956], [−0.189624, 0.03846086, −0.17703873, …, 0.06055506, −0.34405658, −0.03180493]] | Random[0,1] |
| Values of bias in the memory cell and each gate | [0.288068231, 0.591661602, …, 0.881674013, 0.169572993] | Random [0,1] |
| Activation function | Tanh | Sigmoid |

**Table 7** The difference between the actual and predict values results from IFCsAP

| Prediction | Real | Prediction | Real | Prediction | Real | Prediction | Real | Prediction | Real |
|---|---|---|---|---|---|---|---|---|---|
| 34.78846 | 34 | 1.654528 | 6 | 0.412588 | 2 | 50.41773 | 55 | 5.379044 | 5 |
| 36.38553 | 30 | 3.649623 | 3 | 2.326372 | 3 | 53.51553 | 56 | 4.743274 | 2 |
| 31.60161 | 25 | 2.1188 | 5 | 2.995598 | 17 | 55.01352 | 61 | 1.813982 | 5 |
| 26.55325 | 24 | 3.294672 | 5 | 14.30497 | 15 | 60.96044 | 73 | 3.000923 | 4 |
| 24.87279 | 24 | 3.630794 | 5 | 13.25146 | 20 | 73.41774 | 83 | 3.21601 | 4 |
| 25.24641 | 15 | 4.681115 | 7 | 18.33691 | 17 | 83.69789 | 83 | 3.298047 | 3 |
| 15.45882 | 9 | 5.680358 | 3 | 14.43699 | 11 | 83.75974 | 95 | 2.078313 | 4 |
| 9.5449 | 29 | 3.437556 | 4 | 9.669064 | 16 | 95.12534 | 87 | 2.111361 | 3 |
| 28.41048 | 14 | 2.975373 | 1 | 15.17606 | 18 | 87.58269 | 78 | 2.043313 | 5 |
| 16.5466 | 22 | 1.773691 | 9 | 18.48347 | 22 | 78.78084 | 65 | 3.205268 | 4 |
| 24.23592 | 22 | 10.19336 | 6 | 22.88976 | 23 | 65.98693 | 69 | 3.030475 | 4 |
| 25.24196 | 23 | 8.201811 | 5 | 24.72486 | 26 | 69.61816 | 70 | 2.987229 | 4 |
| 25.29594 | 32 | 6.642761 | 7 | 27.65177 | 29 | 71.06481 | 75 | 2.978535 | 2 |
| 34.24964 | 42 | 7.6741 | 6 | 30.22101 | 25 | 75.3206 | 70 | 1.952308 | 0 |
| 44.80105 | 39 | 7.13583 | 2 | 26.2059 | 29 | 69.69443 | 80 | 0.756842 | 3 |
| 42.0159 | 23 | 1.330396 | 0 | 29.46177 | 23 | 78.8698 | 90 | 2.196755 | 3 |
| 23.99235 | 27 | 0.381249 | 4 | 22.86221 | 23 | 89.99757 | 81 | 2.22314 | 7 |
| 28.96297 | 26 | 2.939949 | 3 | 23.30639 | 26 | 81.68404 | 85 | 4.945067 | 6 |
| 28.50837 | 28 | 3.553466 | 4 | 26.34386 | 32 | 84.06477 | 91 | 3.985456 | 7 |
| 31.03599 | 41 | 3.609404 | 3 | 32.18259 | 24 | 89.88517 | 95 | 4.544375 | 3 |
| 44.94321 | 42 | 3.079682 | 4 | 23.9963 | 18 | 93.37846 | 92 | 2.595146 | 4 |
| 45.6139 | 45 | 4.033346 | 5 | 17.69336 | 18 | 91.19576 | 96 | 4.050315 | 3 |
| 48.64455 | 49 | 3.851621 | 3 | 17.61674 | 21 | 94.44313 | 82 | 4.066912 | 4 |
| 52.2008 | 29 | 2.235847 | 4 | 21.44236 | 22 | 81.30444 | 67 | 3.761506 | 1 |
| 31.99407 | 16 | 2.576082 | 0 | 22.08579 | 36 | 66.7011 | 58 | 2.310084 | 5 |
| 15.16854 | 5 | 0.817621 | 2 | 36.52622 | 32 | 57.93331 | 30 | 4.68211 | 8 |
| 5.010992 | 6 | 2.833745 | 6 | 33.55631 | 28 | 31.5391 | 10 | 8.813856 | 8 |
| 6.510222 | 3 | 4.550767 | 5 | 29.84326 | 30 | 9.738421 | 8 | 9.872671 | 6 |
| 4.198856 | 5 | 4.972552 | 7 | 32.7983 | 42 | 6.794169 | 5 | 8.40341 | 5 |
| 5.677699 | 3 | 5.791264 | 8 | 43.71338 | 32 | 4.917524 | 4 | 6.784654 | 4 |
| 3.092324 | 2 | 7.366145 | 5 | 33.3443 | 33 | 3.302827 | 5 | 5.644741 | 8 |
| 2.134996 | 2 | 4.873405 | 4 | 34.66435 | 36 | 3.593681 | 10 | 8.241902 | 15 |
| 2.781274 | 5 | 3.555573 | 2 | 38.07176 | 35 | 6.502742 | 5 | 15.23663 | 15 |
| 4.230802 | 5 | 2.621237 | 2 | 36.56181 | 37 | 3.220106 | 4 | 16.4319 | 16 |
| 4.488424 | 5 | 2.093416 | 0 | 38.63866 | 36 | 2.988119 | 8 | 18.73105 | 14 |
| 3.974105 | 3 | 0.965981 | 1 | 37.45735 | 38 | 5.080526 | 7 | 16.41976 | 14 |
| 2.544927 | 3 | 0.803582 | 0 | 39.6337 | 46 | 4.939282 | 5 | 16.81065 | 16 |
| 2.262641 | 4 | -0.09835 | 0 | 46.94368 | 47 | 4.804131 | 9 | 20.46926 | 30 |
| 2.646403 | 2 | 0.366863 | 0 | 47.38444 | 51 | 8.219869 | 6 | 32.6177 | 20 |
| 20.58939 | 15 | … | … | 2.75937 | 3 | … | … | 13.14352 | 3 |
| 14.81529 | 15 | … | … | 2.542484 | 4 | … | … | 5.996005 | 2 |
| 13.59985 | 15 | … | … | 3.10115 | 5 | … | … | 4.986648 | 3 |
| 13.1015 | 19 | … | … | 3.68614 | 2 | … | … | 4.329667 | 5 |
| 18.86283 | 28 | … | … | 3.029632 | 3 | … | … | 4.925473 | 7 |
| 30.94416 | 35 | … | … | 3.647518 | 1 | … | … | 6.279548 | 7 |
| 37.03983 | 26 | … | … | 2.466094 | 5 | … | … | 6.513453 | 8 |
| 28.33052 | 32 | … | … | 5.201499 | 4 | … | … | 7.796238 | 7 |
| 34.42123 | 35 | … | … | 3.394094 | 7 | … | … | 6.190157 | 5 |
| 37.83168 | 35 | … | … | 5.388619 | 3 | … | … | 4.798513 | 5 |
| 37.46481 | 35 | … | … | 2.641306 | 2 | … | … | 4.556856 | 5 |

**Table 7** continued

| Prediction | Real | Prediction | Real | Prediction | Real | Prediction | Real | Prediction | Real |
|---|---|---|---|---|---|---|---|---|---|
| 37.58238 | 34 | … | … | 2.734388 | 2 | … | … | 4.461099 | 8 |
| 36.37764 | 11 | … | … | 2.734388 | 2 | … | … | 6.501226 | 6 |
| 10.23542 | 9 | … | … | 2.734388 | 4 | … | … | 5.729152 | 12 |
| 8.121566 | 7 | … | … | 4.790839 | 4 | … | … | 9.008903 | 7 |
| 6.889425 | 9 | … | … | 4.790839 | 4 | … | … | 6.113952 | 7 |
| 7.617704 | 8 | … | … | 4.790839 | 3 | … | … | 6.130835 | 7 |
| 6.952367 | 9 | … | … | 3.052802 | 3 | … | … | 6.884548 | 9 |
| 6.659528 | 6 | … | … | 3.052802 | 3 | … | … | 8.843928 | 7 |
| 5.295295 | 19 | … | … | 3.052802 | 5 | … | … | 7.665394 | 5 |
| 15.6793 | 5 | … | … | 5.437168 | 5 | … | … | 8.618543 | 3 |
| 4.46329 | 3 | … | … | 5.437168 | 5 | … | … | 6.016447 | 3 |
| 3.284891 | 4 | … | … | 5.437168 | 3 | … | … | 4.275283 | 5 |
| 3.44795 | 5 | … | … | 3.98441 | 3 | … | … | 5.054852 | 3 |
| 3.93688 | 5 | … | … | 3.98441 | 3 | … | … | 4.867091 | 3 |
| 3.832801 | 6 | … | … | 3.98441 | 3 | … | … | 5.389075 | 6 |
| 4.107695 | 7 | … | … | 3.630408 | 3 | … | … | 6.002992 | 10 |
| 4.831041 | 1 | … | … | 3.630408 | 3 | … | … | 8.479732 | 12 |
| 1.740836 | 1 | … | … | 3.630408 | 4 | … | … | 11.1085 | 14 |
| 1.953708 | 6 | … | … | 5.40837 | 4 | … | … | 13.16331 | 15 |
| 7.004609 | 8 | … | … | 5.40837 | 4 | … | … | 16.6575 | 11 |
| 6.897416 | 6 | … | … | 5.40837 | 7 | … | … | 9.834305 | 11 |
| 6.147143 | 10 | … | … | 8.283098 | 7 | … | … | 9.502816 | 10 |
| 8.156958 | 5 | … | … | 8.283098 | 7 | … | … | 8.462688 | 12 |
| 6.251345 | 3 | … | … | 8.283098 | 15 | … | … | 9.795081 | 18 |
| 5.074443 | 4 | … | … | 15.73539 | 15 | … | … | 15.50082 | 16 |
| 4.809223 | 2 | … | … | 15.73539 | 15 | … | … | 14.20437 | 16 |
| 4.110103 | 5 | … | … | 15.73539 | 16 | … | … | 15.19242 | 17 |
| 6.547019 | 6 | … | … | 14.8975 | 16 | … | … | 17.55351 | 19 |
| 6.262792 | 3 | … | … | 14.8975 | 16 | … | … | 20.86999 | 26 |
| 4.403266 | 1 | … | … | 14.8975 | 30 | … | … | 28.61033 | 35 |
| 3.218214 | 6 | … | … | 30.47908 | 30 | … | … | 38.3054 | 45 |
| 6.983045 | 8 | … | … | 30.47908 | 30 | … | … | 48.51104 | 47 |

## 7 Conclusions

We can summarize the main point performance in that paper as the follows: Building an integrated platform based on physical and program entities in the form of an integrated station ((H/W, S/W) used for essential needs only and reduces the damage resulting from air pollution; thus, this platform saves effort and cost through sensor programming and activating its role to read data on concentrations that cause pollution real-time air, increasing performance, reducing effort, reducing time and cost. Building a special station to measure the concentrations that cause air pollution, which depends on the principle of

**Table 8** SMAPE evaluation

| Name of station | Best spilt | Average lest 25 day | | | | Average SMAPE |
|---|---|---|---|---|---|---|
| Aotizhongxin_Aq | 80%–20% | 0.275028 | | | | 0.275028 |
| Badaling_Aq | 90%–10% | 0.213261 | | | | 0.213261 |
| Beibuxinqu_Aq | 90%–10% | 0.389191 | | | | 0.389191 |
| Daxing_Aq | 80%–20% | 0.250382 | | | | 0.250382 |
| Dingling_Aq | 90%–10% | 0.503017 | | | | 0.503017 |
| Donggaocun_Aq | 70%–30% | 0.371363 | | | | 0.371363 |
| Dongsi_Aq | 50%–50% | 0.221675 | 0.243422 | 0.248539 | 0.255356 | 0.322998 |
| Dongsihuan_Aq | 80%–20% | 0.20021 | | | | 0.20021 |
| Fangshan_Aq | 90%–10% | 0.327049 | | | | 0.327049 |
| Fengtaihuayuan_Aq | 90%–10% | 0.415505 | | | | 0.415505 |
| Guanyuan_Aq | 50%–50% | 0.252592 | 0.266669 | 0.435113 | 0.290471 | 0.414948 |
| Gucheng_Aq | 70%–30% | 0.24183 | 0.218693 | | | 0.230262 |
| Huairou_Aq | 50%–50% | 0.267197 | 0.389582 | 0.360311 | | 0.33903 |
| Liulihe_Aq | 90%–10% | 0.359926 | | | | 0.359926 |
| Mentougou_Aq | 80%–20% | 0.235393 | | | | 0.235393 |
| Miyun_Aq | 60%–40% | 0.145556 | 0.390907 | 0.334637 | | 0.290367 |
| Miyunshuiku_Aq | 80%–20% | 0.43973 | | | | 0.43973 |
| Nansanhuan_Aq | 60%–40% | 0.008667 | 0.231882 | | | 0.120275 |
| Nongzhanguan_Aq | 80%–20% | 0.258566 | | | | 0.258566 |
| Pingchang_Aq | 90%–10% | 0.38889 | | | | 0.38889 |
| Pinggu_Aq | 80%–20% | 0.277051 | | | | 0.277051 |
| Qianmen_Aq | 50%–50% | 0.225878 | 0.300921 | 0.342506 | | 0.289768 |
| Shunyi_Aq | 50%–50% | 0.270642 | 0.277362 | 0.31922 | 0.352798 | 0.406674 |
| Tiantan_Aq | #50%–50% | 0.254859 | 0.272096 | 0.37626 | 0.343996 | 0.415737 |
| Tongzhou_Aq | 80%–20% | 0.244239 | | | | 0.244239 |
| Wanliu_Aq | #60%–40% | 0.278961 | 0.151714 | 0.419545 | | 0.182264 |
| Wanshouxigong_Aq | 80%–20% | 0.273609 | | | | 0.273609 |
| Xizhimenbei_Aq | 90%–10% | 0.273067 | | | | 0.273067 |
| Yanqin_Aq | 80%–20% | 0.235258 | | | | 0.235258 |
| Yizhuang_Aq | 60%–40% | 0.028439 | 0.493526 | 0.252461 | | 0.159835 |
| Yongdingmennei_Aq | 50%–50% | 0.238054 | 0.270327 | 0.262902 | | 0.257095 |
| Yongledian_Aq | 90%–10% | 0.438046 | | | | 0.438046 |
| Yufa_Aq | 80%–20% | 0.267881 | | | | 0.267881 |
| Yungang_Aq | 70%–30% | 0.232745 | 0.199965 | | | 0.216355 |
| Zhiwuyuan_Aq | 50%–50% | 0.31876 | 0.507681 | | | 0.275481 |

Intelligent Data Analysis (IDA). Where data are collected from the stations which are considered as Class Node by the wireless network that was built represented (LoRa & Waspmate) on the calculator which is considered as Master Node. The IFCsAP is fed with the data collected in real time and the preliminary processing is performed on it, after which the predictor results are evaluated using a symmetric mean absolute percentage error (SMAPE). Through this scale, we will evaluate the levels of PM2.5, PM10, NO2, CO and O3. And SO2 for the next 48 h for each station. Often the data contain a proportion of missing or incomplete data, which causes an increase in the prediction or classification error of that data. Therefore, this problem can be addressed by deleting the entries that

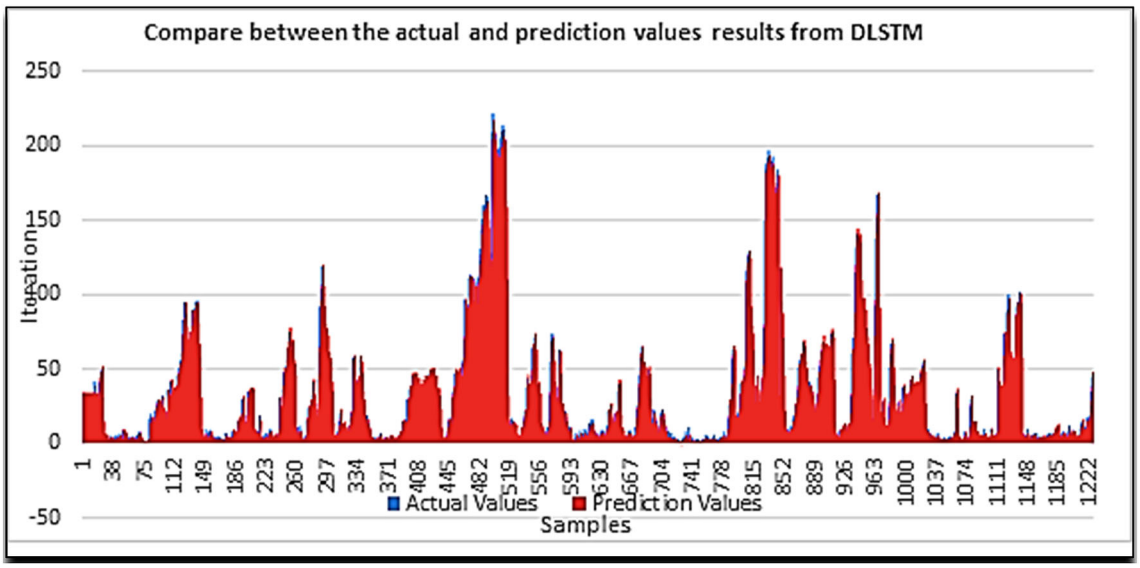


**Fig. 9** Shown actual and prediction values for the first station

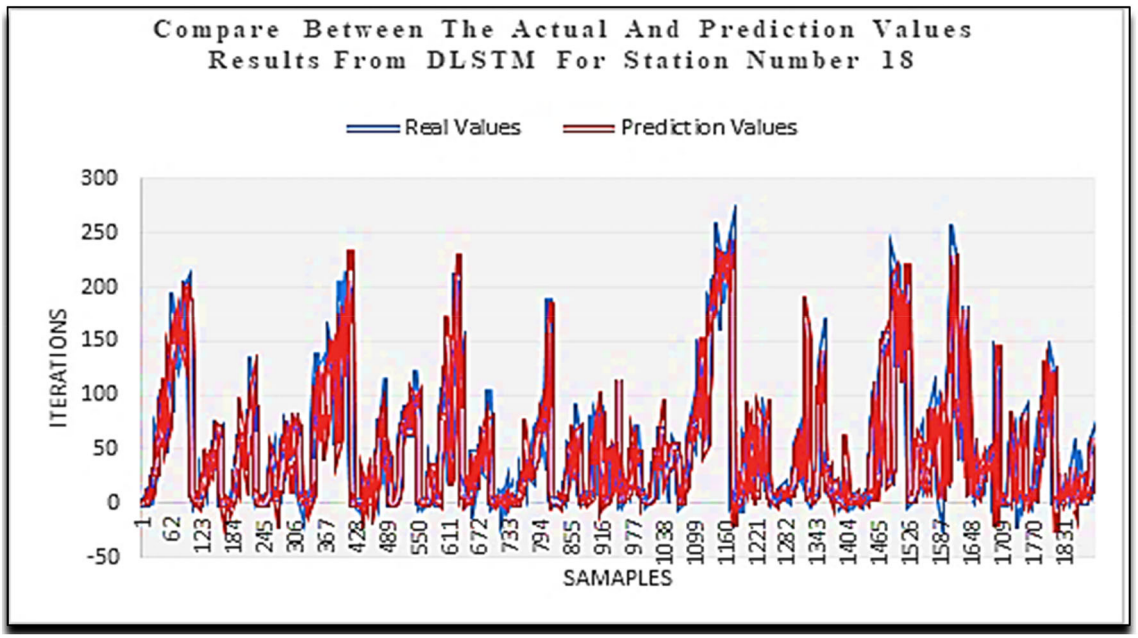


**Fig. 10** Shown actual and prediction values for Station Number 18

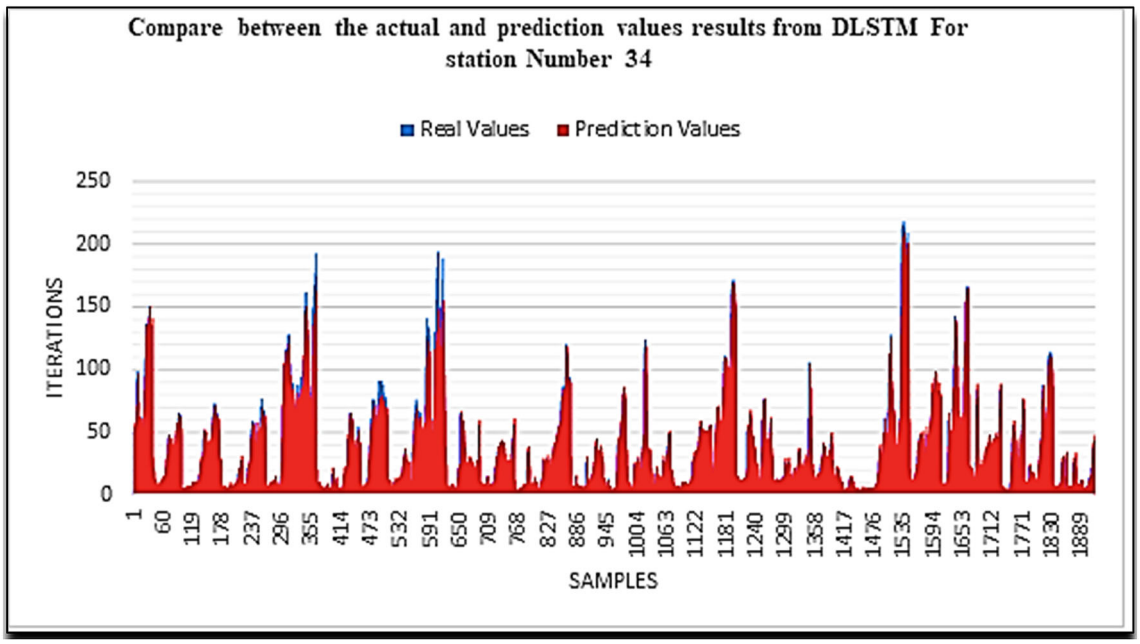


**Fig. 11** Shown actual and prediction values for Station Number 34

contain that data to create a more accurate forecast. The purpose of the Normalization process is to convert data within a specified range of values to be dealt with more accurately in the subsequent stages of processing. In our work, the data were converted within the range [0, 1], because the activation function deals with data within that range. The designed IFCsAP to dealt with one of the most important problems facing the environment at the present time as a result of increased pollution due to electronic waste, factories and laboratories, and the lack of real projects in Iraq to reduce air pollution rates. The designed model proved its accuracy and efficiency in predicting the concentrations that cause air pollution. The designed model is distinguished by the construction of a new tool called DLSTM, which is characterized by its ability to deal with large-size data as well as containing memory that enables it to retain data for long periods. Experiments have shown that the combination of the two technologies that have been designed, which are both DLSTM and DSN-PS, achieve more accurate results and reduce implementation time. The first tool that was designed, DSN-PS, used it to select the best parameters (parameters) to determine the structure of the second tool that was built called DLSTM, thus improving the performance of deep learning models and resulting in the production of a IFCsAP predictor that displays more accurate and efficient results.

The following point gives good idea for features works; Through explore the PSO, to tune other LSTM parameters such as: the learning rate, max error and the number of epochs, instead of based on trial-and-error principles that take too long to find the optimal parameters for the LSTM network. PSO with other deep learning models to find the

**Table 9** Compare SMAPE between the traditional LSTM and IFCsAP

| Name of station | SMAPE of traditional LSTM | SMAPE of IFCsAP |
|---|---|---|
| Aotizhongxin_Aq | 0.386716 | 0.275028 |
| Badaling_Aq | 0.23147 | 0.213261 |
| Beibuxinqu_Aq | 0.721064 | 0.387191 |
| Daxing_Aq | 0.50419 | 0.25038 |
| Dingling_Aq | 0.50749 | 0.503017 |
| Donggaocun_Aq | 0.50846 | 0.371363 |
| Dongsi_Aq | 0.44601 | 0.322998 |
| Dongsihuan_Aq | 0.26456 | 0.20021 |
| Fangshan_Aq | 0.40249 | 0.327049 |
| Fengtaihuayuan_Aq | 0.83241 | 0.415505 |
| Guanyuan_Aq | 0.80799 | 0.414948 |
| Gucheng_Aq | 0.32947 | 0.230262 |
| Huairou_Aq | 0.46668 | 0.33903 |
| Liulihe_Aq | 0.41120 | 0.359926 |
| Mentougou_Aq | 0.32229 | 0.235393 |
| Miyun_Aq | 0.31023 | 0.290367 |
| Miyunshuiku_Aq | 0.55767 | 0.43973 |
| Nansanhuan_Aq | 0.24794 | 0.120275 |
| Nongzhanguan_Aq | 0.35385 | 0.258566 |
| Pingchang_Aq | 0.43291 | 0.38889 |
| Pinggu_Aq | 0.50706 | 0.277051 |
| Qianmen_Aq | 0.33930 | 0.289768 |
| Shunyi_Aq | 0.44912 | 0.406674 |
| Tiantan_Aq | 0.509151 | 0.415737 |
| Tongzhou_Aq | 0.32800 | 0.244239 |
| Wanliu_Aq | 0.45925 | 0.182264 |
| Wanshouxigong_Aq | 0.27640 | 0.273609 |
| Xizhimenbei_Aq | 0.31788 | 0.273067 |
| Yanqin_Aq | 0.49895 | 0.235258 |
| Yizhuang_Aq | 0.66446 | 0.159835 |
| Yongdingmennei_Aq | 0.55448 | 0.257095 |
| Yongledian_Aq | 0.80791 | 0.438046 |
| Yufa_Aq | 0.32257 | 0.267881 |
| Yungang_Aq | 0.28951 | 0.216355 |
| Zhiwuyuan_Aq | 0.56176 | 0.275481 |

best parameters and activation functions, instead of based on trial-and-error principles to find the optimal of network structure. Other type of swarm optimization such as (ant colony optimization, cuckoo search algorithm and glowworm swarm optimization) or a genetic algorithm can be used to find the best parameters and activation functions for LSTM.

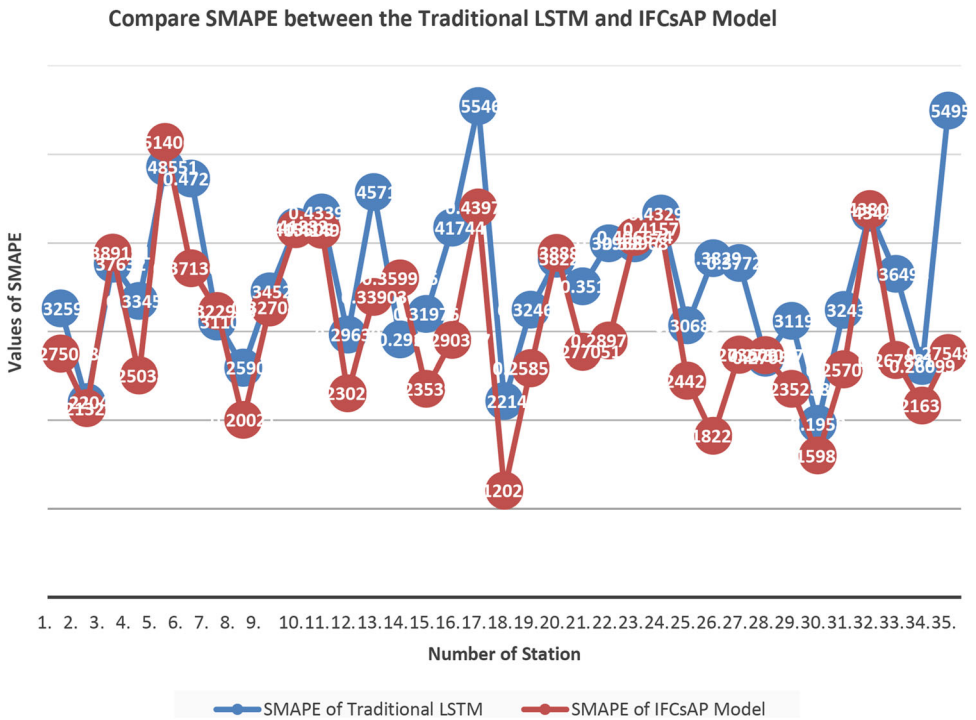


**Fig. 12** Compare SMAPE between the traditional LSTM and IFCsAP model

## Appendix

### Main steps for training LSTM–RNN

In this section, we will show the main steps required to take the decision based on LSTM-RNN, also show how can update variables.

**Step 1**: The forward components

## Step 1.1: Compute the gates

Memory cell:

$$a_t = \tanh(W_a \cdot X_t + U_a \cdot \text{out}_{t-1} + b_a) \quad (1)$$

Input gate:

$$i_t = \sigma(WI \cdot X_t + U_i \cdot \text{out}_{t-1} + b_i) \quad (2)$$

Forget gate:

$$f_t = \sigma\left(Wf \cdot X_t + U_f \cdot \text{out}_{t-1} + b_f\right) \quad (3)$$

Output gate:

$$o_t = \sigma(Wo \cdot X_t + U_o \cdot \text{out}_{t-1} + b_o) \quad (4)$$

Then fined:

Internal state:

$$\text{State}_t = a_t \odot i_t + f_t \odot \text{state}_{t-1} \quad (5)$$

Output:

$$\text{out}_t = \tanh(\text{state}) \odot \text{ot} \quad (6)$$

where

$$\text{Gate } S_t = \begin{bmatrix} a_t \\ i_t \\ f_t \\ o_t \end{bmatrix}, \quad W = \begin{bmatrix} W_a \\ W_i \\ W_f \\ W_o \end{bmatrix}, \quad U = \begin{bmatrix} U_a \\ U_i \\ U_f \\ U_o \end{bmatrix}, \quad b = \begin{bmatrix} b_a \\ b_i \\ b_f \\ b_o \end{bmatrix}$$

**Step. 2**: The backward components:

**Step 2.1. Find**

$\Delta t$ the output difference as computed by any subsequent.

$\Delta$OUT the output difference as computed by the next time-step

$$\delta\text{out}_t = \Delta_t + \Delta\text{out}_t \quad (7)$$

$$\delta\text{state}_t = \delta\text{out}_t \odot o_t \odot \left(1 - \tanh^2(\text{state}_t)\right) + \delta\text{state}_{t+1} \odot f_{t+1} \quad (8)$$

**Step 2.2: Gives**

$$\delta a_t = \delta\text{state}_t \odot i_t \odot \left(1 - a_t^2\right) \quad (9)$$

$$\delta i_t = \delta\text{state}_t \odot a_t \odot i_t \odot (1 - i_t) \quad (10)$$

$$\delta f_t = \delta \text{state}_t \odot \text{state}_{t-1} \odot f_t \odot (1 - f_t) \tag{11}$$

$$\delta o_t = \delta \text{out}_t \odot \tanh(\text{state}_t) \odot o_t \odot (1 - o_t) \tag{12}$$

$$\delta x_t = W^t \cdot \delta \text{state}_t \tag{13}$$

$$\delta \text{out}_{t-1} = U^t \cdot \delta \text{state}_t \tag{14}$$

**Step 3:** update to the internal parameter

$$\delta W = \sum_{t=0}^{T} \delta \text{gates}_t \otimes x_t \tag{15}$$

$$\delta U = \sum_{t=0}^{T} \delta \text{gates}_{t+1} \otimes \text{out}_t \tag{16}$$

$$\delta b = \sum_{t=0}^{T} \delta \text{gates}_{t+1} \tag{17}$$

## Example of LSTM

This example will show calculations performed in LSTM.

Assume the out internal weights:

$$W_a = \begin{bmatrix} 0.45 \\ 0.25 \end{bmatrix}, \quad U_a = [0.15], \quad b_a = [0.2]$$

$$W_i = \begin{bmatrix} 0.95 \\ 0.8 \end{bmatrix}, \quad U_i = [0.8], \quad b_i = [0.65]$$

$$W_f = \begin{bmatrix} 0.7 \\ 0.45 \end{bmatrix}, \quad U_f = [0.1], \quad b_f = [0.15]$$

$$W_o = \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix}, \quad U_o = [0.25], \quad b_o = [0.1]$$

Let input data:

$$X_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{ with label : 0.5}$$

$$X_1 = \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} \text{ with label: 1.25}$$

**Forward t = 0**

$$a_0 = \tanh(W_a * x_0 + U_a * \text{out}_{-1} + b_a = \tanh\left([0.45 \quad 0.25]\begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.15][0] + [0.2]\right) = 0.81775$$

$$i_0 = \sigma(W_i * x_0 + U_i * \text{out}_{-1} + b_i = \sigma\left([0.95 \quad 0.8]\begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.8][0] + [0.65]\right) = 0.96083$$

$$f_0 = \sigma(W_f * x_0 + U_f * \text{out}_{-1} + b_f = \sigma\left([0.7 \quad 0.45]\begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.1][0] + [0.15]\right) = 0.85195$$

$$o_0 = \sigma(W_o * x_0 + U_o * \text{out}_{-1} + b_o = \sigma\left([0.6 \quad 0.4]\begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.25][0] + [0.1]\right) = 0.81757$$

$$\text{state}_0 = a_0 * i_0 + f_0 * \text{state}_{-1} = 0.81775 * 0.96083 + 0.85195 * 0 = 0.78572$$

$$\text{out}_0 = \tanh(\text{state}_0) * o_0 = \tanh(0.78572) * 0.81757 = 0.53631$$

$$a_1 = \tanh(W_a * x_1 + U_a * \text{out}_0 + b_a = \tanh\left([0.45 \quad 0.25]\begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.15][0.53631] + [0.2]\right) = 0.84980$$

$$i_1 = \sigma(W_i * x_1 + U_i * \text{out}_0 + b_i = \sigma\left([0.95 \quad 0.8]\begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.8][0.53631] + [0.65]\right) = 0.98118$$

$$f_1 = \sigma(W_f * x_1 + U_f * \text{out}_0 + b_f = \sigma\left([0.7 \quad 0.45]\begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.1][0.53631] + [0.15]\right) = 0.87030$$

$$o_1 = \sigma(W_o * x_1 + U_o * \text{out}_0 + b_o = \sigma\left([0.6 \quad 0.4]\begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.25][0.53631] + [0.1]\right) = 0.84993$$

$$\text{state}_1 = a_1 * i_1 + f_1 * \text{state}_0 = 0.84980 * 0.98118 + 0.87030 * 0.78572 = 1.5176$$

$$\text{out}_1 = \tanh(\text{state}_1) * o_1 = \tanh(1.5176) * 0.84993 = 0.77197$$

**Backward t = 1**

Compute the different between outputs

$$\partial_x E = x - \hat{x}$$

So, $\Delta_1 = \partial_x E = 0.77197 - 1.25 = -0.47803$.

$\Delta \text{out}_1 = 0$ Because there is no future time – steps

$$\delta \text{out}_1 = \Delta_1 + \Delta \text{out}_1 = -0.47803 + 0 = -0.47803$$

$$\delta \text{state}_1 = \delta \text{out}_1 * o_1 * \left(1 - \tanh^2(\text{state}_1)\right) + \delta \text{state}_2 * f_2 = -0.47803 * 0.84993 * \left(1 - \tanh^2(1.5176)\right) + 0 * 0 = -0.07111$$

$$\delta a_1 = \delta \text{state}_1 * i_1 * \left(1 - a_1^2\right) = -0.07111 * 0.98118 * \left(1 - 0.84680^2\right) = -0.01938$$

$$\delta i_1 = \delta \text{state}_1 * a_1 * i_1 * (1 - i_1) = -0.07111 * 0.84980 * 0.98118 * (1 - 0.98118) = -0.00631$$

$$\begin{aligned}\delta f_1 &= \delta \text{state}_1 * \text{state}_0 * f_1 * (1 - f_1)\\ &= -0.07111 * 0.78572 * 0.78030 * (1 - 0.78030)\\ &= -0.00631\end{aligned}$$

$$\begin{aligned}\delta o_1 &= \delta \text{out}_1 * \tanh(\text{state}_1) * o_1 * (1 - o_1)\\ &= -0.47803 * \tanh(1.5176) * 0.84993\\ &\quad * (1 - 0.84993)\\ &= -0.05538\end{aligned}$$

$$\begin{aligned}\delta x_1 &= W^T * \delta \text{gates}_1\\ &= \begin{bmatrix} 0.45 & 0.95 & 0.70 & 0.60 \\ 0.25 & 0.80 & 0.45 & 0.40 \end{bmatrix} \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix}\\ &= \begin{bmatrix} -0.04743 \\ -0.03073 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}\Delta \text{out}_0 &= U^T * \delta \text{gates}_1\\ &= [\,0.15 \quad 0.80 \quad 0.10 \quad 0.25\,] \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix}\\ &= -0.01828\end{aligned}$$

At this time need return to back our Δout and forwarding on calculation.

**Backward t = 0**

$$\Delta_0 = \partial_x E = 0.53631 - 0.5 = -0.03631$$

$$\Delta \text{out}_1 = -0.01828, \text{ passed back from } T = 1$$

$$\delta \text{out}_0 = \Delta_0 + \Delta \text{out}_0 = 0.03631 + (-0.01828) = 0.01803$$

$$\begin{aligned}\delta \text{state}_0 &= \delta \text{out}_0 * o_0 * \left(1 - \tanh^2(\text{state}_0)\right) + \delta \text{state}_1 * f_1\\ &= -0.01803 * 0.81757 * \left(1 - \tanh^2(0.78572)\right)\\ &\quad + (-0.07111 * 0.87030) = -0.05349\end{aligned}$$

$$\begin{aligned}\delta a_0 &= \delta \text{state}_0 * i_0 * \left(1 - a_0^2\right)\\ &= -0.05349 * 0.96083 * (1 - 0.81775) = -0.01703\end{aligned}$$

$$\begin{aligned}\delta i_0 &= \delta \text{state}_0 * a_0 * i_0 * (1 - i_0)\\ &= -0.05349 * 0.81775 * 0.96083 * (1 - 0.96083)\\ &= -0.00165\end{aligned}$$

$$\begin{aligned}\delta f_0 &= \delta \text{state}_0 * \text{state}_{-1} * f_0 * (1 - f_0)\\ &= -0.05349 * 0 * 0.85195 * (1 - 0.85195) = 0\end{aligned}$$

$$\begin{aligned}\delta o_0 &= \delta \text{out}_0 * \tanh(\text{state}_0) * o_0 * (1 - o_0)\\ &= 0.01803 * \tanh(0.78572) * 0.81757 * (1 - 0.81757)\\ &= 0.00176\end{aligned}$$

$$\begin{aligned}\delta x_0 &= W^T * \delta \text{gates}_0\\ &= \begin{bmatrix} 0.45 & 0.95 & 0.70 & 0.60 \\ 0.25 & 0.80 & 0.45 & 0.40 \end{bmatrix} \begin{bmatrix} -0.01703 \\ -0.00165 \\ 0 \\ 0.00176 \end{bmatrix}\\ &= \begin{bmatrix} -0.00817 \\ -0.00487 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}\Delta \text{out}_{-1} &= U^T * \delta \text{gates}_1\\ &= [\,0.15 \quad 0.80 \quad 0.10 \quad 0.25\,] \begin{bmatrix} -0.01703 \\ -0.00165 \\ 0 \\ 0.00176 \end{bmatrix}\\ &= -0.00343\end{aligned}$$

After complete backward, let $\lambda = 0.1$ and update the parameters by:

$$\begin{aligned}\delta W &= \sum_{t=0}^{T} \delta \text{gates}_t \otimes X_t\\ &= \begin{bmatrix} -0.01703 \\ -0.00165 \\ 0 \\ 0.00176 \end{bmatrix} [1.02.0] + \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix} [0.53.0]\\ &= \begin{bmatrix} -0.02672 & -0.0922 \\ -0.00221 & -0.00666 \\ -0.00316 & -0.01893 \\ -0.02593 & -0.16262 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}\delta U &= \sum_{t=0}^{T} \delta \text{gates}_{t+1} \otimes \text{out}_t = \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix} [0.53631]\\ &= \begin{bmatrix} -0.01039 \\ -0.00060 \\ -0.00338 \\ -0.02970 \end{bmatrix}\end{aligned}$$

$$\begin{aligned}\delta b &= \sum_{t=0}^{T} \delta \text{gates}_{t+1} = \begin{bmatrix} -0.01703 \\ -0.00165 \\ 0 \\ 0.00176 \end{bmatrix} + \begin{bmatrix} -0.01938 \\ -0.00112 \\ -0.00631 \\ -0.05538 \end{bmatrix}\\ &= \begin{bmatrix} -0.03641 \\ -0.00277 \\ -0.00631 \\ -0.05362 \end{bmatrix}\end{aligned}$$

Depend on the SGD updating out parameters:

$$W^{\text{new}} = W^{\text{old}} - \lambda * \delta W^{\text{old}}$$

$$W_a = \begin{bmatrix} 0.45267 \\ 0.25922 \end{bmatrix}, \quad U_a = [0.15104], \quad b_a = [0.20364].$$

$$W_i = \begin{bmatrix} 0.95022 \\ 0.80067 \end{bmatrix}, \quad U_i = [0.80006], \quad b_i = [0.65028]$$

$$W_f = \begin{bmatrix} 0.70031 \\ 0.45189 \end{bmatrix}, \quad U_f = [0.10034], \quad b_f = [0.15063]$$

$$W_o = \begin{bmatrix} 0.60259 \\ 0.41626 \end{bmatrix}, \quad U_o = [0.25297], \quad b_o = [0.10536]$$

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the author.

## References


1. Al-Janabi S, Alkaim AF, Adel Z (2020) An innovative synthesis of deep learning techniques (DCapsNet and DCOM) for generation electrical renewable energy from wind energy. Soft Comput 24:10943–10962. https://doi.org/10.1007/s00500-020-04905-9
2. Alkaim AF, Al_Janabi S (2020) Multi objectives optimization to gas flaring reduction from oil production. In: Farhaoui Y (ed) Big data and networks technologies. BDNT 2019. Lecture notes in networks andsystems, vol 81. Springer, Cham. https://doi.org/10.1007/978-3-030-23672-4_10
3. Chen B et al (2018) How do people in different places experience different levels of air pollution? Using worldwide Chinese as a lens. Environ Pollut 238:874–883. https://doi.org/10.1016/J.ENVPOL.2018.03.093
4. Donahue NM (2018) Air pollution and air quality. Green Chem. https://doi.org/10.1016/B978-0-12-809270-5.00007-8
5. Das HS, Roy P (2019) A deep dive into deep learning techniques for solving spoken language identification problems. Intell Speech Signal Process. https://doi.org/10.1016/B978-0-12-818130-0.00005-2
6. Aunan K, Hansen MH, Liu Z, Wang S (2019) The hidden hazard of household air pollution in rural China. Environ Sci Policy 93:27–33. https://doi.org/10.1016/J.ENVSCI.2018.12.004
7. Basavaraju S, Gaj S, Sur A (2019) Object memorability prediction using deep learning: location and size bias. J Vis Commun Image Represent 59:117–127. https://doi.org/10.1016/J.JVCIR.2019.01.008
8. Al-Janabi S, Alkaim AF (2020) A nifty collaborative analysis to predicting a novel tool (DRFLLS) for missing values estimation. Soft Comput 24(1):555–569. https://doi.org/10.1007/s00500-019-03972-x
9. Al-Janabi S, Mahdi MA (2019) Evaluation prediction techniques to achievement an optimal biomedical analysis. Int J Grid Util Comput 10(5):512–527
10. Chien J-T, Chien J-T (2019) Deep neural network. Source Sep Mach Learn. https://doi.org/10.1016/B978-0-12-804566-4.00019-X
11. Al-Janabi S, Alkaim AF (2021) A comparative analysis of DNA protein synthesis for solving optimization problems: a novel nature-inspired algorithm. In: Abraham A, Sasaki H, Rios R, Gandhi N, Singh U, Ma K (eds) Innovations in bio-inspired computing and applications. IBICA 2020. Advances in Intelligent Systems and Computing, vol 1372. Springer, Cham. https://doi.org/10.1007/978-3-030-73603-3_1
12. Congcong W, Shufu L, Xiaojing Y, Ling P, Xiang L, Yuan H, Tianhe C (2019) A novel spatiotemporal convolutional long short-term neural network for air pollution prediction. J Sci Total Environ 654:1091–1099
13. Al-Janabi S et al (2015) Design and evaluation of a hybrid system for detection and prediction of faults in electrical transformers. Int J Elect Power Energy Syst 67:324–335. https://doi.org/10.1016/j.ijepes.2014.12.005
14. Bianchi FM, Maiorino E, Kampffmeyer MC et al (2017) An overview and comparative analysis of recurrent neural networks for short term load forecasting. https://doi.org/10.1007/978-3-319-70338-1
15. Al-Janabi S, Salman AH (2021) "Sensitive integration of multilevel optimization model in human activity recognition for smartphone and smartwatch applications". In: Big data mining and analytics, vol 4, no 2, pp 124–138. https://doi.org/10.26599/BDMA.2020.9020022
16. Samaher AJ (2020) Smart system to create an optimal higher education environment using IDA and IOTs. Int J Comput Appl 42(3):244–259. https://doi.org/10.1080/1206212X.2018.1512460
17. Al-Janabi S, Al-Shourbaji I, Shojafar M, Abdelhag M (2017) Mobile cloud computing: challenges and future research directions. In: 2017 10th international conference on developments in esystems engineering (DeSE), pp 62–67. https://doi.org/10.1109/DeSE.2017.21
18. Tebrean B, Crisan S, Muresan C, Crisan TE (2017) Low cost command and control system for automated infusion devices. In: Vlad S, Roman N (eds) International conference on advancements of medicineand health care through technology; 12th–15th October 2016, Cluj-Napoca, Romania. IFMBE Proceedings, vol 59. Springer, Cham. https://doi.org/10.1007/978-3-319-52875-5_18
19. Li X, Peng L, Yao X, Cui S, Hu Y, You C, Chi T (2017) Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation. EnvironPollut 231(Pt 1):997–1004. https://doi.org/10.1016/j.envpol.2017.08.114
20. Wu L, Li N, Yang Y (2018) Prediction of air quality indicators for the Beijing-Tianjin-Hebei region. J Clean Prod 196:682–687. https://doi.org/10.1016/j.jclepro.2018.06.068
21. Wen C et al (2019) A novel spatiotemporal convolutional long short-term neural network for air pollution prediction. Sci Total Environ 654:1091–1099. https://doi.org/10.1016/j.scitotenv.2018.11.086
22. Shang Z, Deng T, He J, Duan X (2019) A novel model for hourly PM2.5 concentration prediction based on CART and EELM. Sci Total Environ 651:3043–3052. https://doi.org/10.1016/j.scitotenv.2018.10.193
23. Li H, Wang J, Li R, Lu H (2019) Novel analysis–forecast system based on multi-objective optimization for air quality index. J Clean Prod 208:1365–1383. https://doi.org/10.1016/j.jclepro.2018.10.129
24. Kim SY, Bechle M, Hankey S, Sheppard L, Szpiro AA, Marshall JD (2020) Concentrations of criteria pollutants in the contiguous US, 1979–2015: role of prediction model parsimony in integrated empirical geographic regression. PLoS ONE. https://doi.org/10.1371/journal.pone.0228535
25. Matos J, Faria RPV, Nogueira IBR et al (2019) Optimization strategies for chiral separation by true moving bed chromatography using Particles Swarm Optimization (PSO) and new

Parallel PSO variant. Comput Chem Eng 123:344–356. https://doi.org/10.1016/J.COMPCHEMENG.2019.01.020

26. Al-Janabi S, Alwan E (2017) Soft mathematical system to solve black box problem through development the FARB based on hyperbolic and polynomial functions. In: 2017 10th international conference on developments in esystems engineering (DeSE), pp 37–42. https://doi.org/10.1109/DeSE.2017.23
27. Zhou B-Z, Liu X-F, Cai G-P et al (2019) Motion prediction of an uncontrolled space target. Adv Sp Res 63:496–511. https://doi.org/10.1016/J.ASR.2018.09.025