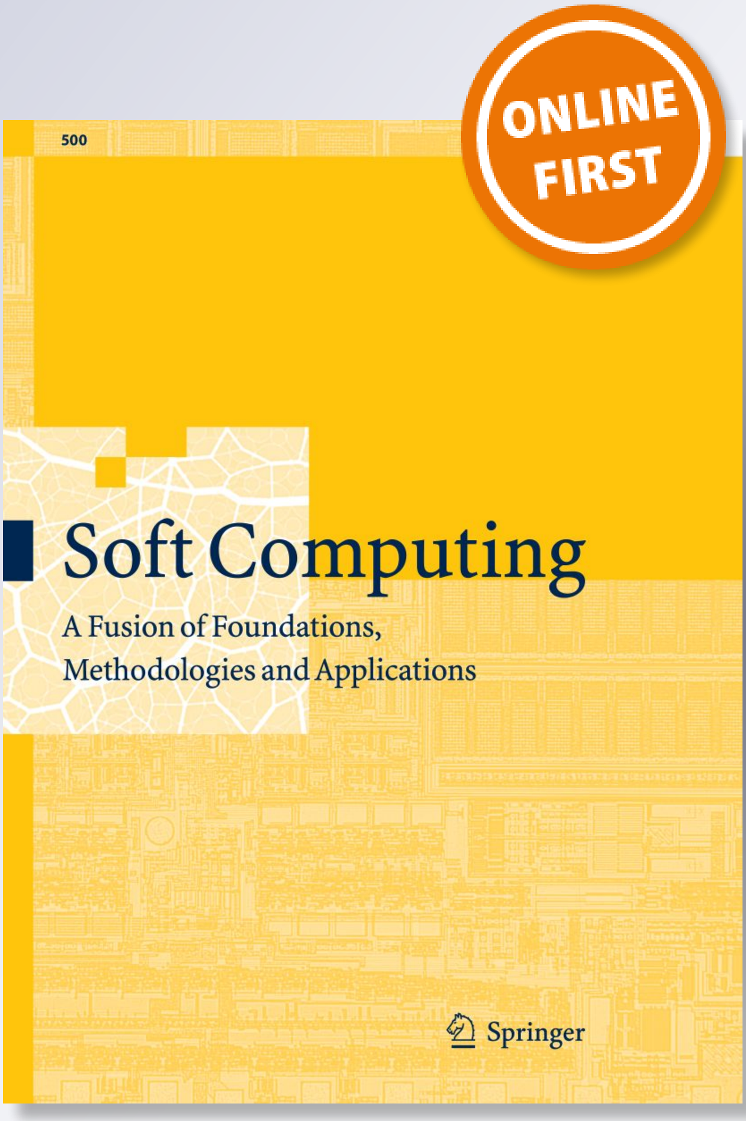
Samaher Al-Janabi, Mustafa Mohammad & Ali Al-Sultan

Soft Computing
A Fusion of Foundations,
Methodologies and Applications

ISSN 1432-7643

Soft Comput
DOI 10.1007/s00500-019-04495-1



500

ONLINE FIRST

Soft Computing

A Fusion of Foundations,
Methodologies and Applications

Springer

Soft Computing
https://doi.org/10.1007/s00500-019-04495-1

METHODOLOGIES AND APPLICATION

A new method for prediction of air pollution based on intelligent computation

Samaher Al-Janabi[1] · Mustafa Mohammad[1] · Ali Al-Sultan[1]

Abstract

The detection and treatment of increasing air pollution due to technological developments represent some of the most important challenges facing the world today. Indeed, there has been a significant increase in levels of environmental pollution in recent years. The aim of the work presented herein is to design an intelligent predictor for the concentrations of air pollutants over the next 2 days based on deep learning techniques using a recurrent neural network (RNN). The best structure for its operation is then determined using a particle swarm optimization (PSO) algorithm. The new predictor based on intelligent computation relying on unsupervised learning, i.e., long short-term memory (LSTM) and optimization (i.e., PSO), is called the smart air quality prediction model (SAQPM). The main goal is to predict six the concentrations of six types of air pollution, viz. PM2.5 particulate matter, PM10, particulate matter, nitrogen dioxide (NO_2), carbon monoxide (CO), ozone (O_3), and sulfur dioxide (SO_2). SAQPM consists of four stages. The first stage involves data collection from multiple stations (35 in this case). The second stage involves preprocessing of the data, including (a) separation of each station with an independent focus, (b) handle missing values, and (c) normalization of the dataset to the range of (0, 1) using the MinMaxScalar method. The third stage relates to building the predictor based on the LSTM method by identifying the best structure and parameter values (weight, bias, number of hidden layers, number of nodes in each hidden layer, and activation function) for the network using the functional PSO algorithm to achieve a goal. Thereafter, the dataset is split into training and testing parts based on the ten cross-validation principle. The training dataset is then used to build the predictor. In the fourth stage, evaluation results for each station are obtained by reading the concentration of each pollutant each hour for at most 30 days then taking the average of the symmetric mean absolute percentage error (SMAPE) for 25 days only.

Keywords Air pollutants · Big data · Prediction · Analytical solution · Long short-term memory · Particle swarm algorithm · Intelligent computation

1 Introduction

Data is one of the most valuable treasures in the world, forming the basis of different branches of computer science. Data refers to any set of objects with organized features, or specific characteristics of an object or collection thereof and their features. Data can be of different types and can be obtained by observation, search, or recording (Alkaim & Al-Janabi 2020). In general, researchers dealing with the concept called data science work in three domains, related to data, intelligence, and statistics (Buyya et al. 2016). Data science can be divided into three fields, viz. small, normal, and big/huge data. Small data is organized into uniform structures such as tables or lists containing no more than 30 samples and thus

Communicated by V. Loia.

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

Mustafa Mohammad
musmmus505@gmail.com

Ali Al-Sultan
wsci.ali.yakoob@uobabylon.edu.iq

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

Published online: 28 November 2019

does not follow the normal distribution and cannot be used for decision-making. On the other hand, normal data is also structured but does follow the normal distribution and is thus useful for taking different types of decision such as clustering, classification, prediction, optimization, etc. Finally, big data can have different types such as structured, semistructured, or unstructured, with size ranging from 1 TB to 1 ZB. Extraction of useful knowledge or patterns from big data can be achieved by the combination of the two main concepts of machine learning and cloud computing.

Deep learning is a branch of modern science that considers multilevel learning processes, where learning is applied at each level for a specific part of the problem and aggregation of the corresponding results enables the overall problem to be solved. It is thus classified as a branch of artificial intelligence (Liu et al. 2019).

Prediction is a type of decision-making technique where future events are forecast based on historical information. Among the three types of prediction technique, viz. traditional (offering accuracy), self (offering speed), and intelligent (offering both speed and accuracy), this work relates to the latter (Al-Janabi et al. 2015).

Increasing air pollution caused by technological development represents one of the most important challenges facing the world today. It can be categorized into several classes depending on its origin, viz. pollution due to living organisms such as bacteria and fungi in the environment such as water, air, or soil; chemical air pollution due to an the imbalance in the ecosystem resulting from chemical effects, being in the form of solid particles or liquid droplets or gases; and technological, due to a change in the balance between the components of an ecosystem that prevents its efficient operation and ability to perform its natural role in the disposal of pollutants.

2 Related work

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.

Ong et al. (2015) used a deep recurrent neural network (DRNN) reinforced with a novel pretraining system using an autoencoder, principally designed for time-series prediction. Moreover, the sensors were chosen within the DRNN without degrading the accuracy of the predictions by considering the sparsity of the system. This method was applied to the prediction of air pollution, in particular for PM2.5 particulate matter concentration, offering more accurate results compared with the poor performance achieved using the noise reduction approach. The results were evaluated using four measures, viz. the root-mean-square error (RMSE), precision (*P*), recall (*R*), and *F* measure. The work presented herein is similar in that it uses the same technique (RNN), albeit based on the LSTM approach.

Al-Janabi et al. (2015) applied a hybrid system using genetic neural computing (GNC) to analyze and understand data corresponding to the concentration of dissolved gases in four subgroups for analysis based on the IEEE C57.104 specification using a genetic algorithm (GA). The clustering data was input to the neural network to predict the different types of errors. The hybrid system generates decision rules which identify the error accurately. Two measures were used in that work, viz. the Davies–Bouldin (DB) index and the mean square error (MSE). The results indicated that the problem could be solved at lower cost and that the described method facilitated the prediction process and enabled a more accurate approach through the analysis of errors and ways to address them. This work is similar to that presented herein in that it uses neural networks, while the difference lies in the use of the PSO algorithm combined with LSTM.

Li et al. (2016) described an air quality prediction method based on a spatiotemporal deep learning (STDL) model. A stacked autoencoder (SAE) method was applied to extract inherent air quality characteristics, being trained using a greedy layerwise method. In comparison with traditional time-series prediction models, the described model could predict the air quality at all stations at the same time and exhibited temporal stability across all seasons. In addition, a comparison with the spatiotemporal artificial neural network (STANN), autoregression moving average (ARMA), and support vector regression (SVR) models was presented. The results of the model were evaluated using three measures, viz. RMSE, mean absolute error (MAE), and mean absolute percentage error (MAPE). The work presented herein is similar in that the same technique (RNN) is applied to prediction the air quality indexes, but now dealing with huge data and also applying the LSTM approach to enhance the operation of the network.

Li et al. (2017) 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.

Ghoneim and Manjunatha (2017) described a new prediction model based on deep learning for ozone levels, considering pollution and weather correlations in an integrated fashion. This deep learning model was used to learn ozone level features, and trained using a grid search technique. A deep architecture model is utilized to represent the ozone level features for the predictions. Experiments demonstrated that the proposed method offered superior performance for ozone level predictions. The results of this study could be helpful for predicting ozone level pollution in Aarhus City as a model for smart cities, to improve the accuracy of ozone forecasting tools. The results of the model were evaluated based on the RMSE, MAE, MAPE, squared R^2, and correlation coefficient. The work presented herein also uses a memory (LSTM in this case) for processing of large data, but differs in that the optimal structure of the neural network is found by applying a PSO algorithm.

Lifeng et al. (2018) 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, SO_2, NO_2, 8-h O_3, 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, SO_2, NO_2, O_8–O_3, and O_3 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.

Popoola et al. (2018) considered sensor measurements including SNAQ boxes and network deployment, sensor measurement validation, and source apportionment to build a predictive model for the ADMS-Airport tool, using the concentration of pollutants to determine the air quality model. The results showed that such a method can be applied in many environments that suffer from air pollution, potentially reducing the health effects of reduced air quality and decreasing cost, as well as for monitoring of greenhouse-gas emissions. The work presented herein is similar in that the concentration of air pollutants is determined, but differs in its use of the LSTM RNN method.

For effective extraction of spatiotemporal features, Wen et al. (2019) 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. (2019) 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, O_3, PM10, SO_2, NO_2, CO) but differs in terms of the mechanism applied to reduce air pollutants, applying the RNN method.

Li et al. (2019) 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.

Table 1 Comparison of previous works

Name	Dataset/database	Method	Evaluation	Advantage
Ong et al. (2015)	Air quality index (AQI) http://uk-air.defra.gov.uk	DRNN enhanced with a novel pretraining method using an autoencoder	RMSE *P* *R* F	(1) Numerical experiments showed that use of the DRNN with the proposed pretraining method was superior to canonical and state-of-the-art autoencoder training methods for time-series prediction, compared with the VENUS PM2.5 prediction system; (2) NN (known as RNN, in contrast to FNN) showed very good performance for modeling temporal structures and was successfully applied to many real-world problems
Al-Janabi et al. (2015)		GNC BPNN	MSE (DB)	This work showed that the problem could be solved at lower cost and that the method facilitates the prediction process and provides greater accuracy based on the analysis of errors and ways to address them
Li et al. (2016)	Air quality using PM2.5 http://datacenter.mep.gov.cn/	STDL-based air quality prediction method SAE	RMSE MAE MAPE	In contrast to traditional prediction models for air quality time series, the model predicted the air quality at all monitoring stations simultaneously and showed satisfactory seasonal stability. The performance of the proposed method was compared with that of the STANN, ARMA, and SVR models, revealing that it was effective and outperformed these competitors
Li et al. (2017)	Air quality using PM2.5 http://datacenter.mep.gov.cn/	LSTM	RMSE MAE MAPE	The model is capable of modeling time series with long time dependencies and can automatically determine the optimum time lags. The performance of the model was compared with that of six other models, including the LSTME approach considered herein, traditional LSTM NN, STDL, TDNN, ARMA, and SVR
Ghoneim and Manjunatha (2017)	CityPulse dataset http://iot.ee.surrey.ac.uk	(1) Deep learning approach (2) In-memory computing	RMSE MAE MAPE R^2 *r*	The method was evaluated on the CityPluse dataset and compared with SVM, NN, and a generalized linear model (GLM). Comparison of the results revealed that the proposed model is efficient and superior compared with existing models
Lifeng et al. (2018)	*Beijing Smog: when Growth Trumps Life in China* (BBC, 2013) www.bbc.com/news/magazine-21198265	FGM(1,1)	RMSE MAPE	The FGM (1.1) method showed much higher performance than the traditional GM model (1.1), revealing that the average annual concentrations of PM2.5, PM10, SO_2, NO_2, O_8–O_3, and O_3 24-h will decrease
Popoola et al. (2018)		SNAQ boxes and network deployment Sensor measurement validation Source apportionment		The results of this study revealed that this method could be applied in many environments that suffer from air pollution

Table 1 (continued)

Name	Dataset/database	Method	Evaluation	Advantage
Wen et al. (2019)	Hourly PM2.5 concentration data from 1233 air quality monitoring stations in China collected from 1 January 2016 to 31 December 2017, acquired from the Ministry of Environmental Protection of China http://datacenter.mep.gov.cn/	Combination of CNN and LSTM-NN	RMSE MAE MAPE	(1) Addition of PM2.5 information from neighboring stations, which contributes to the spatiality of the data, considerably improved the prediction accuracy of the model (2) Use of auxiliary data helped predict sudden changes in air quality, thereby improving the prediction performance of the model. Moreover, compared with meteorological data, aerosol optical depth (AOD) data contributed more to the accuracy of the model (3) The model could efficiently extract more essential spatiotemporal correlation features based on the combination of 3D CNN and stateful LSTM, thereby achieving air quality predictions with greater accuracy and stability over different spatiotemporal scales
Shang et al. (2019).	Yancheng City, 1 of the 13 cities under the direct administration of Jiangsu Province, China, ranging from 32°34′–34°28′ N, 119°27′–120°54′ E	CART and EELM	RMSE MAPE	The experimental results revealed that the method could address the global–local duplication of the prediction method at each leaf and that CART–EELM work better compared with the RF, v-SVR, and EELM models, as well as showing superior performance compared with the EELM and *k*-means-EELM seasonal approaches
Li et al. (2019)	Datasets from eight sites in China https://data.epmap.org	LSSVM MOMV	RMSE MAE MAPE	Application of the proposed method to the dataset revealed good performance for analysis and control of air quality, as well as the approximation of values with high precision

Table 1 presents a comparison of the cited previous works based on the type of dataset considered, the methodology used, the evaluation measures applied, and the advantages offered.

3 Main concept

3.1 Big data

This term is commonly used today due to the abundance and diversity of sources, which lead to difficulty in dealing with the resulting data because they may be unorganized and require large storage systems. Big data was first defined by Douglas Laney based on the 3Vs, viz. volume, velocity, and variety, and has been widely cited since 2001, although many have tried to increase the number of Vs, to 4, 5, 6, and even 11. Big data has also been defined as an application, which emphasizes its different applications based on the different types of data, whereas Barry Devlin defined it as the application of process-mediated data, human-sourced information, and machine-generated data. Shaun Connolly focused on analyzing the transactions, interactions, and observation of data, seeking hindsights using big data technology; this type of definition is oriented by new technological developments such as MapReduce, bulk synchronous parallel (BSP) computing such as Hama, resilient distributed datasets (RDDs) such as Spark, and Lambda architecture such as Flink (Buyya et al. 2016) (Fig. 1).

3.2 Big data analysis stages

Data analysis is the process of inspecting, transforming, and modeling data with the goal of discovering useful information (Al-Janabi & Alkaim 2019), informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names while being used in different business, science, and social science domains. In today's business, data analysis plays a role in making decision-making more scientific and helping businesses operate effectively (Buyya et al. 2016) (Fig. 2).

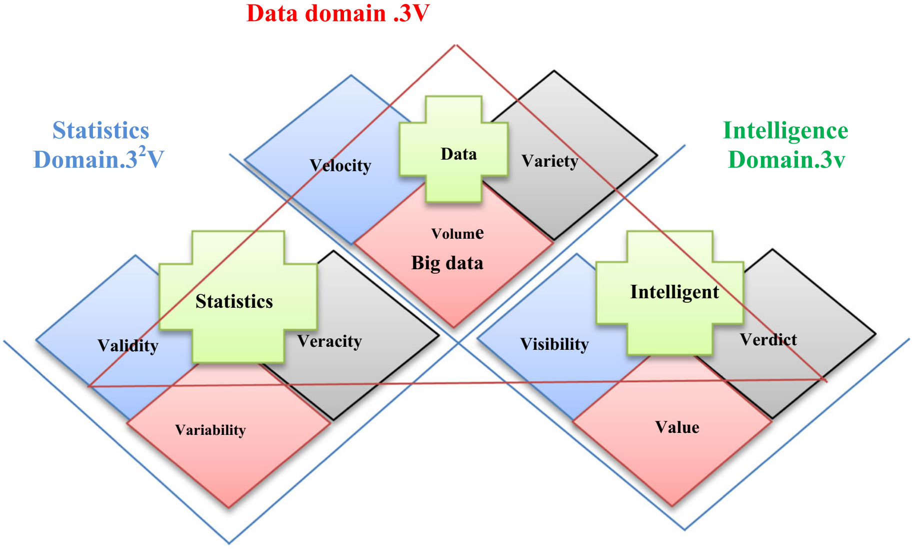


Fig. 1 Big data

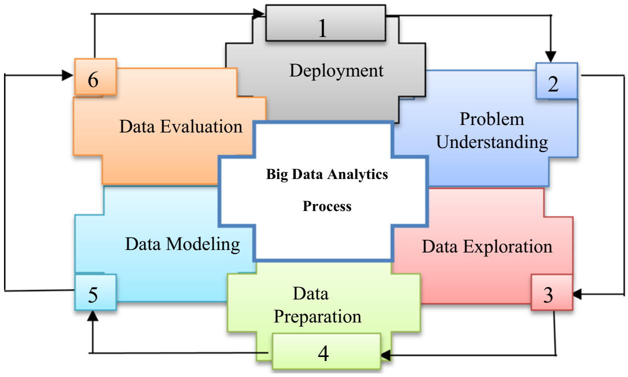


Fig. 2 Big data analysis stages

3.3 Deep learning

Deep learning is a new area of machine learning which has gained popularity in the recent past. Deep learning refers to architectures that contain multiple hidden layers (deep networks) to learn different features at multiple levels of abstraction. Deep learning algorithms seek to exploit the unknown structure in the input distribution to discover useful representations, often at multiple levels, where higher-level learned features are defined in terms of lower-level features (Ali et al. 2019).

3.4 Prediction

Prediction can be defined as the task of data analysis to predict unknown values of target features. It includes a classification task for class label prediction and numerical prediction whose aim is to predict continuous or ordered values. The type of target attribute specifies whether the problem is classification with binary values or numerical prediction with continuous values. Many statistical methodologies have been used for such numerical prediction, among which regression analysis is most often applied (Basavaraju et al. 2019) (Fig. 3).

3.5 Air pollution

Air pollution remains a serious concern and has attracted attention from industries, governments, as well as the scientific community. One type of air pollutant that has attracted immense attention is fine particulate matter. PM2.5 is a widespread air pollutant, consisting of a mixture of solid and liquid particles suspended in the air, in addition to PM10 and O_3 as other types of air pollution. Air pollution is a global issue that transcends geographical boundaries and calls for an interdisciplinary approach to solve a global problem. Thus, forecasting concentrations of air pollutants is an effective method for protecting public health by providing early warnings of harmful air pollutants (Liu et al. 2019).

4 Building the DLSTM-PSO model

In this section, an effective prediction model is built, including four stages. The first stage involves dataset preprocessing, including data collection, splitting, handling of missing values, and normalization of the dataset. In the second stage, the PSO algorithm is applied to identify the best structure for the LSTM network, including determination of the best weight, bias, number of hidden layers, number of nodes in each hidden layer, and activation function. In the third stage, the prediction model (called DLSTM-PSO) is built to predict the concentration of the six pollutants considered. The final stage is the evaluation of the results based on the symmetric mean absolute percentage error (SMAPE) and 10 cross-validations.

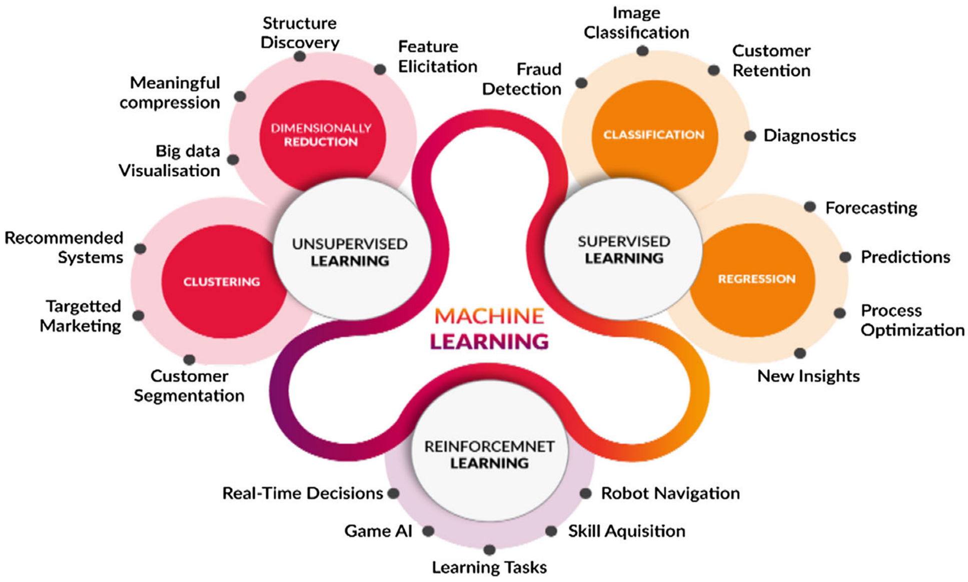


Fig. 3 Main types of machine learning techniques (Al-Janabi & Mahdi 2019)

First, the main details of the model are presented, being based on the following assumptions:

- The air quality data file contains the concentration of several major air pollutants: PM2.5 (μg/m^3), PM10 (μg/m^3), and NO_2 (μg/m^3). In addition, we also provide the concentrations of CO (mg/m^3), O_3 (μg/m^3), and SO_2 (-μg/m^3) from Beijing 2018.
- All points with NA or negative values in the PM2.5 or PM10 (or O_3 from) data are considered to be invalid and are dropped from the truth file; For example, the data [2957631, CT3, 2018-04-13 00:00:00, 24.0,,15. 7,] has no PM2.5 data and thus is dropped from the scoring matrix, even though it includes PM10 data.
- The PM2.5 limit is taken as 10 μg/m^3 (average allowable value per year) or 25 μg/m^3 (average allowable value in 24 h).
- The PM10 limit is taken as 20 μg/m^3 (average allowable value per year) or 50 μg/m^3 (average allowable value).
- The O_3 limit is taken as 100 μg/m^3 (average allowable value in 8 h). The recommended maximum value, previously set at 120 μg/m^3 in 8 h, has been reduced to 100 μg/m^3 based on recent findings of the relationships between daily mortality and ozone levels in locations where the concentration of this substance is less than 120 μg/m^3.
- The NO_2 limit is taken as 40 μg/m^3 (average allowable value per year) or 200 μg/m^3 (average allowable value per hour).
- The SO_2 limit is taken as 20 μg/m^3 (average allowable value in 24 h) or 500 μg/m^3 (average allowable value in 10 min) (Fig. 4).

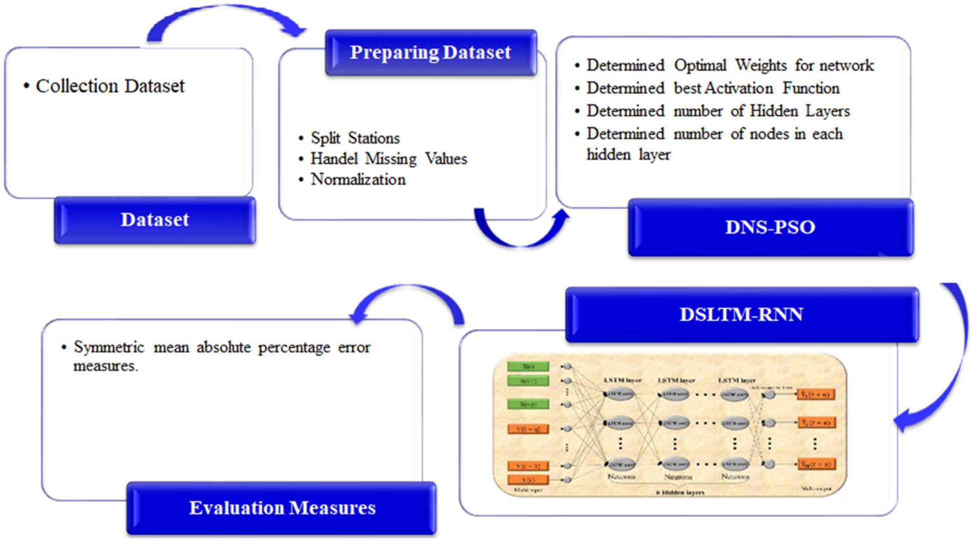

• Collection Dataset
Dataset
Preparing Dataset
• Split Stations
• Handel Missing Values
• Normalization
• Determined Optimal Weights for network
• Determined best Activation Function
• Determined number of Hidden Layers
• Determined number of nodes in each hidden layer
DNS-PSO
DSLTM-RNN
• Symmetric mean absolute percentage error measures.
Evaluation Measures

Fig. 4 Block diagram of proposed DLSTM-RNN

```
Algorithm#1: DLSTM-PSO
Input: Air pollution dataset include multi station and six concentration of
        $PM2_{2.5}$, $PM_{10}$, $NO_2$, $CO$, $O_3$, $SO_2$
Output: Predict the air quality index
// Pre-Processing Stage
1: For each row in air pollution dataset
2:     Call split Stations
3:     For each column in air pollution dataset
4:         Call handle missing value
5:         Call normalize
6:     End for
7: End for
// Build DLSTM-PSO Predictor
8: For each id_station
9:     For i in range (1: total number of samples [id_station]
10:        Split dataset according 10- Cross Validation into Training and Testing
           dataset
11:    End for
12:    For each Training part not used
13:        Call DSN-PS        //determined the structure & parameters of model
14:        Call DLSTM         //predictive the value of six concertation
15:    End for
16:    For each Testing part not used
17:        Test stopping conditions // max number of epoch and max error
                                       generation
18:        IF max error generation < Emax
19:            Go to step 25
20:        Else
21:            GO to step 10
22:        End IF
23:    End for
24: End for
// Evaluation stage
25: Call Evaluation SMAPE
End SAQPM
```

4.1 Data preprocessing stage

As explained above, datasets were collected from two type of resources (i.e., from the directory of websites such as the KDD Cup 2018 dataset, or by building stations to capture concentrations). These datasets must be handling before using them to build predictive models. The dataset for each station was split and saved in a separate file containing the name of the station. Then, missing values were treated by dropping each row from which one or more value was lacking. Finally, normalization was applied in each column of the dataset related to each station, using the MinMaxScaler process on all the dataset (PM2.5, PM10, NO_2, CO, O_3, and SO_2) to make the concentration values lie in the range [0, 1]. The main steps in this stage are described in Algorithm 2.

Algorithm#2: Pre-processing

```
Input: Air pollution dataset include multi station and six concentration of
        $PM2_{2.5}$, $PM_{10}$, $NO_2$, CO, $O_3$, $SO_2$
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 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 = Nan
            droop the row contain 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
    End for
End for
End pre-processing
```

4.2 Determined structure network-particle swarm (DSN-PS)

In this step, the structure of the LSTM is specified by PSO. During the PSO, the three following steps are repeated until the maximum epoch limit is reached or one of the stop conditions is met:

- Calculate the value of the fit for each element among the particles;
- Update the pBest appropriate values for each particle, as well as the best gBest general value;
- Update the speed and position of each particle.

The aim of the PSO algorithm is to optimize the LSTM-RNN by specifying the optimal values of the weight, bias, number of hidden layers, number of nodes in each hidden layer, and activation function, as shown in the diagram in Fig. 5 and presented in Tables 2 and 3.

Algorithm#3: DSN-PS

Input: *Split station [id_station] have clean data in range [0, 1], problem size, Population size*

Output: *Optimal of (weights, bias, number of hidden layer, number of node in each hidden layer, and activation function)*

Initialization*: A: Population of agents, p_i: Position of agent a_i in the solution space, f : Objective functions, v_i : Velocity of agent's a_i , $V(a_i)$: Neighborhood of agent a_i (fixed), W = random weights, B=random bias, H=number of hidden layer >257, N=number of node in each hidden layer>257 , AF= call activation function*

```
For i=1 to it_max
    For number of hhidden layer h in H  do
        fh = f(h)
        IF fh is better than f(hBest)
            hBest = h
        End IF
    End for
    For number of node of each hidden layer n in N do
        fn = f(n)
        IF fn is better than f(nBest)
            nBest = n
        End IF
    End for
    For ranodom weights w in W do
        fw = f(w)
        IF fw is better than f(wBest)
            wBest = w
        End IF
    End for
    For ranodom bias b in B do
        fb = f(b)
        IF fb is better than f(bBest)
            bBest = b
        End IF
    End for
    Call activation function af in AF
        faf = f(af)
        IF faf is better than f(afBest)
            afBest = af
        End IF
    ghBest = best h in H
    For number of hhidden layer h in H  do
        hv = hv + c1*rand*(hBest – h) + c2*rand*(ghBest – h)
        h = h + hv
    End for
    gnBest = best n in N
    For number of node of each hidden layer n in N do
        nv = nv + c1*rand*(nBest – n) + c2*rand*(gnBest – n)
        n = n +nv
    End for
    gwBest = best w in W
    For ranodom weights w in W do
        wv = wv + c1*rand*(wBest – w) + c2*rand*(gwBest – w)
        w = w + wv
    End for

    gbBest = best b in B
    For ranodom bias b in B do
        bv = bv + c1*rand*(bBest – b) + c2*rand*(gbBest – b)
        b = b +bv
    End for
    gafBest = best af in AF
    Call activation function af in AF
        afv = afv + c1*rand*(afBest – af) + c2*rand*(gafBest – af)
        af = af + afv
End for
End DNS-PS
```

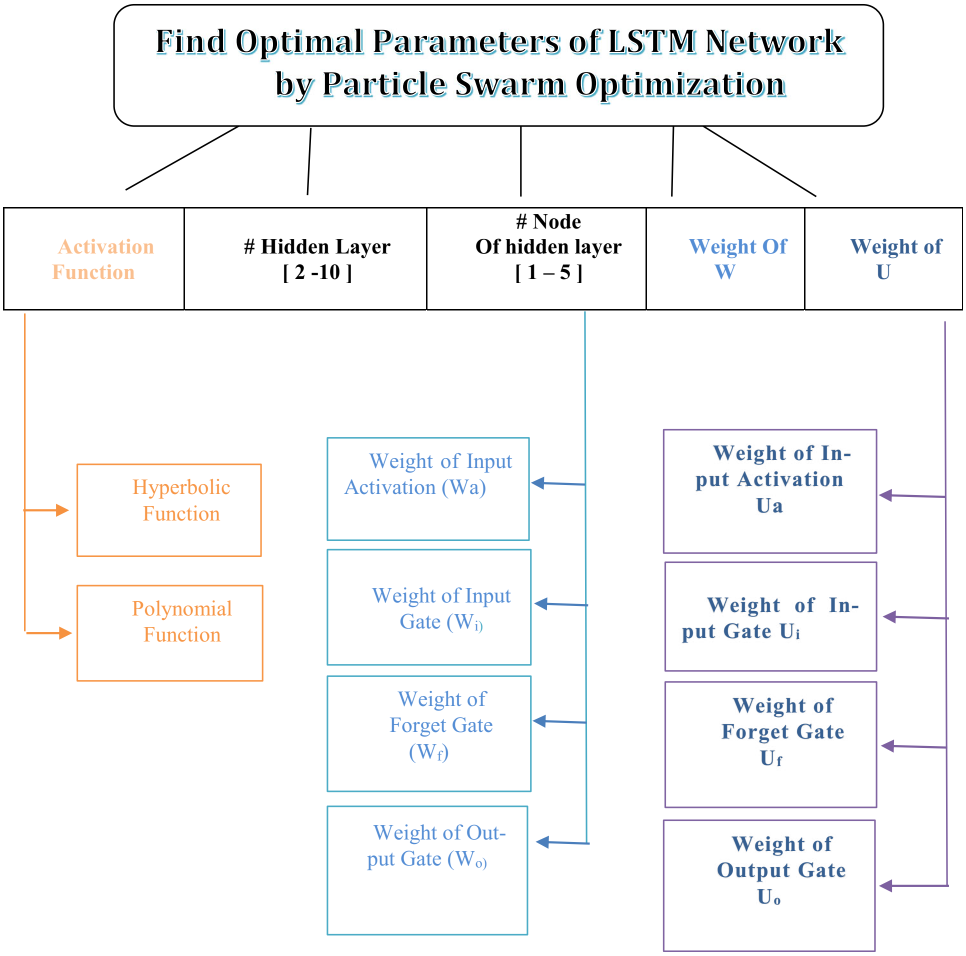

Find Optimal Parameters of LSTM Network by Particle Swarm Optimization
Activation Function
# Hidden Layer [2 -10]
# Node Of hidden layer [1 – 5]
Weight Of W
Weight of U
Hyperbolic Function
Polynomial Function
Weight of Input Activation (Wa)
Weight of Input Gate (Wi)
Weight of Forget Gate (Wf)
Weight of Output Gate (Wo)
Weight of Input Activation Ua
Weight of Input Gate Ui
Weight of Forget Gate Uf
Weight of Output Gate Uo

Fig. 5 Determination of the optimal parameters of the LSTM using PSO

Table 2 Hyperbolic Functions with description (Al-Janabi and Alwan 2017)

Hyperbolic functions		
Name	No. of variables	Function
Sinh	1	$F(x) = \frac{e^x - e^{-x}}{2}$
Cosh	1	$F(x) = \frac{e^x + e^{-x}}{2}$
Tanh	1	$F(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
Sinh^{-1}	1	$F(x) = \frac{2}{e^x - e^{-x}}$
Cosh^{-1}	1	$F(x) = \frac{2}{e^x + e^{-x}}$
Tanh^{-1}	1	$F(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}}$

x is the input, and $F(x)$ is the output

Table 3 Polynomial Functions with description (Al-Janabi and Alwan 2017)

Polynomial function		
Name	No. of variables	Function
Linear	1	$F(x) = p_1 + p_2 \times x_1$
Linear	2	$F(x) = p_1 + p_2 \times x_1 + p_3 \times x_2$
Linear	3	$F(x) = p_1 + p_2 \times x_1 + p_3 \times x_2 + p_4 \times x_3$
Quadratic	1	$F(x) = p_1 + p_2 \times x_1 + p_3 \times x_1^2$
Quadratic	2	$F(x) = 1 + p_2 \times x_1 + p_3 \times x_1^2 + p_4 \times x_2 + p_5 \times x_2^2 + p_6 \times x_1 \times x_2$
Cubic	1	$F(x) = p_1 + p_2 \times x_1 + p_3 \times x_1^2 + p_4 \times x_1^3$
Product	2	$F(x) = p_1 + p_2 \times x_1 \times x_2$
Ratio	2	$F(x) = p_1 + p_2 \times (x_1/x_2)$
Logistic	1	$F(x) = p_1 + p_2/(1 + \exp(p_3 \times (x_1 - p_4)))$
Log	1	$F(x) = p_1 + p_2 \times \mathrm{Log}(x_1 + p_3)$
Exponential	1	$F(x) = p_1 + p_2 \times \exp((p_3 \times (x_1 + p_4))$
Asymptotic	1	$F(x) = p_1 + p_2/(x_1 + p_3)$

x is input

$F(x)$ is output

p_1, p_2, p_3, and p_4 are constants

4.3 Development of the long short-term memory (DLSTM) approach

The common LSTM module consists of a cell, an input port, an output port, and a forgotten gateway. The cell remembers values at random intervals, and the three gates regulate the flow of information inside and outside the cell.

LSTM networks are well suited for classifying, processing, and predicting predictions based on time-series data, where there may be an unknown delay for important events in a time series. LSTMs have been developed to deal with fading problems that can be encountered when training traditional RNNs. The relative lack of sense of the gap length is a feature of LSTM in RNNs, hidden Markov models, and other sequential learning methods in many applications (Inácio et al. 2019).

An LSTM module (or cell) has five essential components, which allows it to model both long- and short-term data:

- *Memory cell*: This represents the internal memory of the cell, which stores both short- and long-term memories.
- *Hidden state*: This is the output state information calculated from the current input, previous hidden state, and current cell input, eventually being used to predict the future concentrations. Additionally, the hidden state can decide to retrieve only the short- or long-term or both types of memory stored in the cell state to make the next prediction.
- *Input gate*: Decides how much information from the current input flows to the cell state.
- *Forget gate*: Decides how much information from the current input and the previous cell state flows into the current cell state.
- *Output gate*: Decides how much information from the current cell state flows into the hidden state.

4.3.1 The variables in LSTM–RNN

This algorithm requires multiple variables to be set at the beginning, then these are updated by applying computational operations, as shown below:

Step 1: The forward components

Step 1.1: Compute the gates:
Input activation:

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

Input gate:

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

Forget gate:

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

Output gate:

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

Then find:
Internal state:

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

Output:

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

where

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

Step 2: The backward components:

Step 2.1: Find
Δt, the output difference as computed by any subsequent
ΔOUT, 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: This gives

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

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

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

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

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

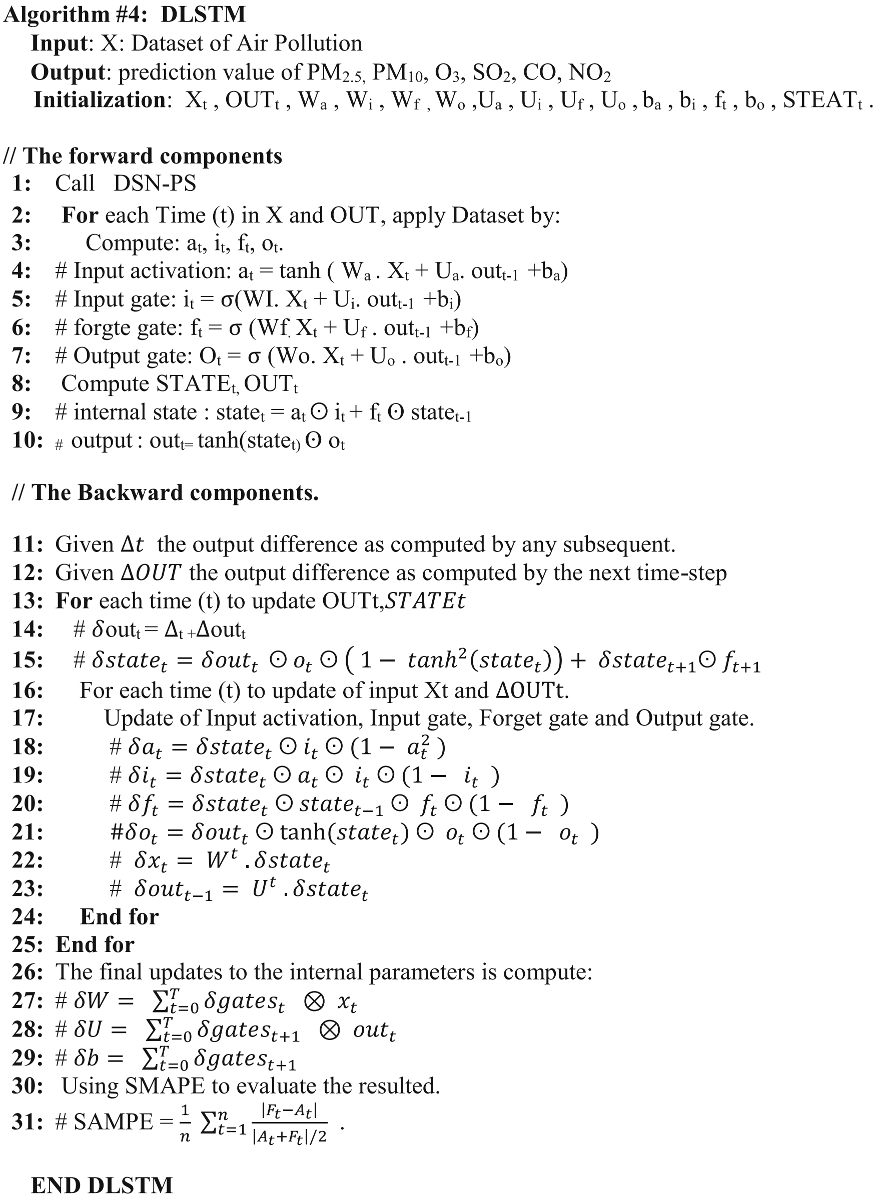
$$\delta \text{out}_{t-1} = U^t . \delta \text{state}_t \tag{14}$$

Step 3: Update the internal parameter.

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

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

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



```
Algorithm #4:  DLSTM
  Input: X: Dataset of Air Pollution
  Output: prediction value of $PM_{2.5,}$ $PM_{10}$, $O_3$, $SO_2$, CO, $NO_2$
  Initialization:  $X_t$ , $OUT_t$ , $W_a$ , $W_i$ , $W_{f\ ,}$ $W_o$ ,$U_a$ , $U_i$ , $U_f$ , $U_o$ , $b_a$ , $b_i$ , $f_t$ , $b_o$ , $STEAT_t$ .

// The forward components
  Call   DSN-PS
   For each Time (t) in X and OUT, apply Dataset by:
      Compute: $a_t$, $i_t$, $f_t$, $o_t$.
  # Input activation: $a_t = \tanh$ ( $W_a$ . $X_t$ + $U_a$. $out_{t-1}$ $+b_a$)
  # Input gate: $i_t = \sigma$(WI. $X_t$ + $U_i$. $out_{t-1}$ $+b_i$)
  # forgte gate: $f_t = \sigma$ (Wf. $X_t$ + $U_f$ . $out_{t-1}$ $+b_f$)
  # Output gate: $O_t = \sigma$ (Wo. $X_t$ + $U_o$ . $out_{t-1}$ $+b_o$)
   Compute $STATE_{t,}$ $OUT_t$
  # internal state : $state_t = a_t \odot i_t + f_t\ \Theta\ state_{t-1}$
  # output : $out_{t=}$ $\tanh(state_{t)}\ \Theta\ o_t$

// The Backward components.

  Given $\Delta t$  the output difference as computed by any subsequent.
  Given $\Delta OUT$ the output difference as computed by the next time-step
  For each time (t) to update OUTt,$STATEt$
    # $\delta out_t = \Delta_{t\ +}\Delta out_t$
    # $\delta state_t = \delta out_t \odot o_t \odot \big(1 - tanh^2(state_t)\big) + \delta state_{t+1} \odot f_{t+1}$
     For each time (t) to update of input Xt and ΔOUTt.
        Update of Input activation, Input gate, Forget gate and Output gate.
         # $\delta a_t = \delta state_t \odot i_t \odot (1 - a_t^2\ )$
         # $\delta i_t = \delta state_t \odot a_t \odot i_t \odot (1 - i_t\ )$
         # $\delta f_t = \delta state_t \odot state_{t-1} \odot f_t \odot (1 - f_t\ )$
         #$\delta o_t = \delta out_t \odot \tanh(state_t) \odot o_t \odot (1 - o_t\ )$
         # $\delta x_t = W^t\ .\delta state_t$
         # $\delta out_{t-1} = U^t\ .\delta state_t$
     End for
  End for
  The final updates to the internal parameters is compute:
  # $\delta W = \sum_{t=0}^{T} \delta gates_t \otimes x_t$
  # $\delta U = \sum_{t=0}^{T} \delta gates_{t+1} \otimes out_t$
  # $\delta b = \sum_{t=0}^{T} \delta gates_{t+1}$
   Using SMAPE to evaluate the resulted.
  # SAMPE = $\frac{1}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{|A_t + F_t|/2}$ .

  END DLSTM
```

5 Experiment

This section presents the results of each stage in the prediction model. Also, a justification is presented for all the results.

5.1 Dataset used

Data from KDD cup 2018* is used, containing the name of the 35 stations and the concentration of each pollutant per hour, viz. PM2.5, PM10, SO_x, CO, NO_2, and O_3. Table 4 presents the raw dataset.

The dataset is split by station and saved in separate files containing the name of each station. Thereby, each station contains six types of concentrations and a number of records as presented in Table 5.

Missing values in each column are then handled as illustrated in Table 6.

Table 7 presents a description of the results after handling missing values.

5.2 Data visualization

Figure 6 illustrates the resulting data, which contains various patterns occurring over time.

This graph already says a lot. The specific reason for picking this data is that this graph presents a wide range of different behaviors in the concentrations of the air pollutants over time, which will make the learning process more robust and provide the opportunity to test the quality of the predictions in a variety of situations.

Another feature to notice is that the values at the beginning of 2017 are much higher and fluctuate more than the values close to the end of the dataset. Therefore, one must ensure that the data exhibit similar value ranges throughout the time frame, which will be considered during the data normalization phase.

5.3 Normalizing the data

Before normalizing, the dataset is split into a training set and a test set, using 70% for training and 30% for testing.

A scaler must now be defined to normalize the data. MinMaxScalar scales all the data to the region of 0 and 1. One can also reshape the training and test data to have the shape [data_size, num_features].

Due to the observation above that different time periods of the data have different value ranges, one should normalize the data after splitting the full series into windows. Otherwise, the earlier data will be close to 0 and will not add much value to the learning process. Here, a window size of 2500 is chosen.

The data can now be smoothed using an exponential moving average, which helps to remove the inherent raggedness of the concentration data and produce a smoother curve. Note that only the training data should be smoothed in this way.

5.4 Data generator

A data generator is first implemented to train our model, including a method called unroll_batches(…) that will output a set of num_unrollings batches of input data obtained sequentially, where each batch of data is of size [batch_size, 1]. Then, each batch of input data will have a corresponding output batch of data (Tables 8, 9).

For example, if num_unrollings = 3 and batch_size = 4, a set of unrolled batches might look like:

- input data: [x_0,$x_1$0, $x_2$0, $x_3$0, $x_4$0, $x_5$0], [x_1,$x_1$1, $x_2$1, $x_3$1, $x_4$1, $x_5$1], [x_2,$x_1$2, $x_2$2, $x_3$2,$x_4$2, $x_3$5]
- output data[x_1,$x_1$1, $x_2$1, $x_3$1, $x_4$1, $x_5$1],[x_2,$x_1$2, $x_2$2, $x_3$2,$x_4$2, $x_5$2], [x_3,$x_1$3, $x_2$3, $x_3$3,$x_4$3, $x_5$3]

Then, one finds the best weights of the input between the hidden layers as illustrated in Tables 10, 11, and 12.

Table 4 Dataset before handling missing values

No.	utc_time	Station	PM2.5	PM10	NO_2	O_3	CO	SO_2
6	1/1/2017 19:00	aotizhongxin_aq	429.0	141.0	6.5	3.0	9.0	NaN
7	1/1/2017 20:00	aotizhongxin_aq	211.0	110.0	3.3	11.0	NaN	NaN
…								
…								
175	1/8/2017 20:00	aotizhongxin_aq	6.0	20.0	0.3	4.0	NaN	NaN
176	1/8/2017 21:00	aotizhongxin_aq	6.0	28.0	0.2	49.0	4.0	NaN
….								
….								

Table 5 Split stations

No.	Name of station	No. of features	No. of records
1.	aotizhongxin_aq	6	8886
2.	badaling_aq	6	8886
3.	beibuxinqu_aq	6	8886
4.	daxing_aq	6	8886
5.	dingling_aq	6	8886
6.	donggaocun_aq	6	8886
7.	dongsi_aq	6	8886
8.	dongsihuan_aq	6	8886
9.	fangshan_aq	6	8886
10.	fengtaihuayuan_aq	6	8886
11.	guanyuan_aq	6	8886
12.	gucheng_aq	6	8886
13.	huairou_aq	6	8886
14.	liulihe_aq	6	8886
15.	mentougou_aq	6	8886
16.	miyun_aq	6	8886
17.	miyunshuiku_aq	6	8886
18.	miyunshuiku_aq	6	8886
19.	nongzhanguan_aq	6	8886
20.	pingchang_aq	6	8886
21.	pinggu_aq	6	8886
22.	qianmen_aq	6	8886
23.	shunyi_aq	6	8886
24.	tiantan_aq	6	8886
25.	tongzhou_aq	6	8886
26.	wanliu_aq	6	8886
27.	wanshouxigong_aq	6	8886
28.	xizhimenbei_aq	6	8886
29.	yanqin_aq	6	8886
30.	yizhuang_aq	6	8886
31.	yongdingmennei_aq	6	8886
32.	yongledian_aq	6	8886
33.	yufa_aq	6	8886
34.	yungang_aq	6	8886
35.	zhiwuyuan_aq	6	8886

Meanwhile, the weights of the recurrent connections are illustrated in Tables 13 and 14.

After constructing the LSTM model using the DSN-PS algorithm, the model consists of several layers capable of predicting the concentrations of the air pollutants. We have 32 stations × 6 pollutants (PM2.5, PM10, SO_x, CO, NO_x, and O_3), resulting in 192 readings per hour, 4608 per day, and 138,240 within the 30 days of the training process of the network. After the training, DLSTM-RNN can predict air pollution concentrations over the next 48 h based on the previous training. The SMAPE error rate scale is then used to evaluate the results from the DLSTM network for the least or nearest error (Table 15).

The combination of LSTM and SPO reduces the training time for the network because the SPO algorithm provides the best function for activation and identifies the number of hidden layers and number of nodes in each hidden layer, considering that they provide better weights, but at the same time complicate the network for the reason described above.

6 Discussion and conclusions

Air quality index datasets represent huge data, requiring intelligent and deep computation for the extraction of useful patterns. Despite the advantage of their large size, the limitations of such datasets include the possibility of missing values, that each concentration may show high and low value, and that the records for each station may not be equal.

DSN-PS is used to determine the parameters and activation function of the DLSTM, offering the advantage of reduced time of execution for LSTM, and limitation of the DSN-PS will increase the complexity of the LSTM.

Table 6 Dataset after handling missing values

No.	utc_time	Station	PM2.5	PM10	NO_2	O_3	CO	SO_2
6	1/1/2017 19:00	aotizhongxin_aq	429.0	141.0	6.5	3.0	9.0	11.212
7	1/1/2017 20:00	aotizhongxin_aq	211.0	110.0	3.3	11.0	15.78	11.212
…								
…								
175	1/8/2017 20:00	aotizhongxin_aq	6.0	20.0	0.3	4.0	15.78	11.212
176	1/8/2017 21:00	aotizhongxin_aq	6.0	28.0	0.2	49.0	4.0	11.212
…								

The DLSTM is constructed by using the LSTM of the DSN-PS, and PSO is used to determine the optimal number of hidden layers, number of nodes in each hidden layer, weight, bias, and activation function. The advantage of

Table 7 Description of data after preprocessing

	PM2.5	PM10	NO_2	CO	O_3	SO_2
Count	200.000	200.000	200.000	200.000	200.000	200.000
Mean	179.949	134.376	27.180	17.205	15.788	11.212
Std	131.835	123.790	56.373	20.671	11.056	2.788
Min	5.000	4.600	0.200	0.200	2.000	2.000
Max	500.000	561.000	208.000	79.000	61.000	37.000

DLSTM is its ability to deal with huge data and the use of memory cells to save information in the long term, while the limitation of DSTM is the huge number of parameters.

Evaluation is the process of calculating the error between actual and predicted values, which can be achieved using different types of error measure, including prediction (i.e., SMAPE, MSE, RMSE, MAE, MAPE, etc.) and coefficient matrixes (i.e., accuracy, *F*, FP, etc.).

- *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 SAQPM 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 beneficial result from building predictor by the combination between DSNPS and DLSTM?*

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

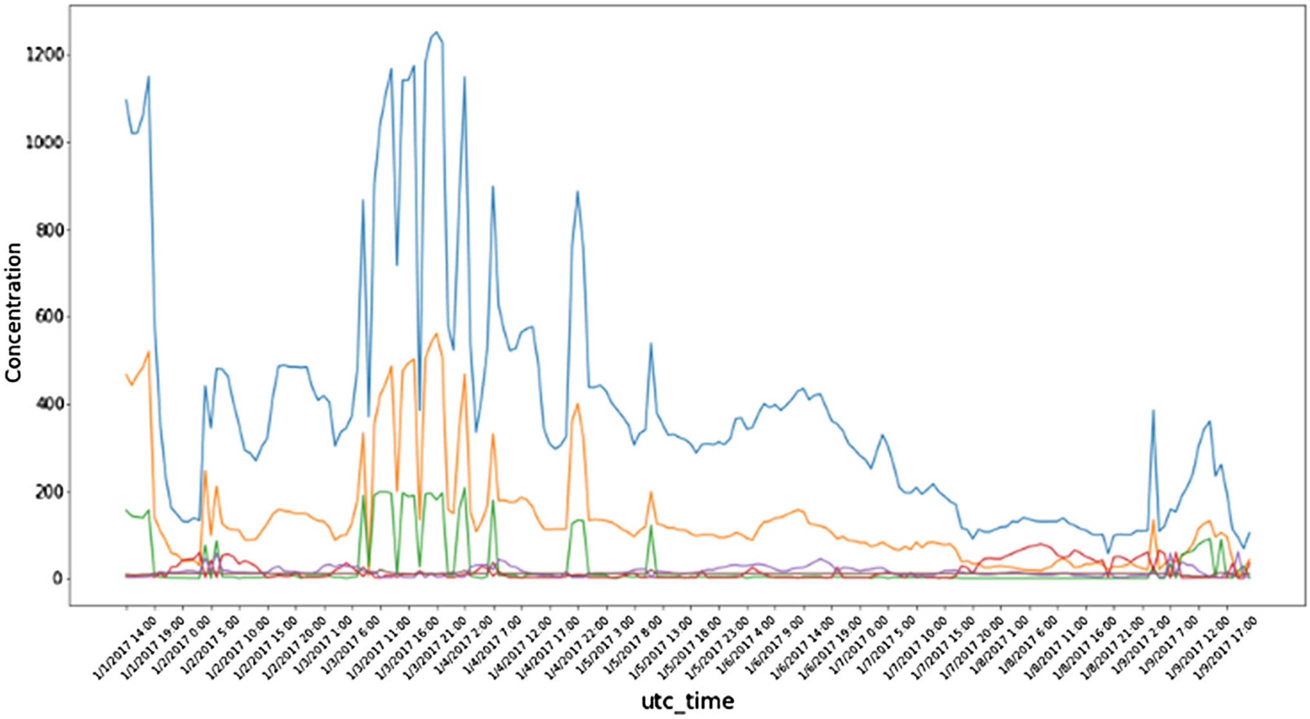


Fig. 6 Data visualization

Table 8 Dataset after normalization and splitting for training and testing

Station		PM2.5		PM10		NO_2	
aotizhongxin_aq	Train	0.63380282	0	0.54107981	0.03508772	0.74634146	0.25892857
	Test	0	0.376	0.07017544	0.45614035	0.25892857	0.75
badaling_aq	Train	0.3032491	0.20454545	0.26201923	0.2745098	0.55944056	0.36567164
	Test	0.22272727	0.16363636	0.29019608	0.43529412	0.39552239	0.43283582
…	…						
	…						
yungang_aq	Train	0.62329803	0.29487179	0.55191874	0.43915344	0.59292035	0.44217687
	Test	0.32692308	0.23717949	0.47619048	0.4021164	0.3877551	0.21768707
zhiwuyuan_aq	Train	0.97430407	0.45454545	0.8627451	0.50549451	0.96875	0.78
	Test	0.58441558	0.03896104	0.61538462	0.08241758	1.08	0.5
Station		CO		O_3		SO_2	
aotizhongxin_aq	Train	0.71	0.04761905	0.00546448	0.28985507	0.04320988	0
	Test	0.04761905	0.38095238	0.27536232	0.05797101	0	0.4
badaling_aq	Train	0.26415094	0.3125	0	0	0.58024691	0
	Test	0.3125	0.375	0	0.05649718	0.03333333	0.53333333
…	…						
	…						
yungang_aq	Train	0.52066116	0.3	0	0.36521739	0.03571429	0.52
	Test	0.35	0.25	0.47826087	0.33043478	0.48	0.48
zhiwuyuan_aq	Train	0.85106383	0.41176471	0.03896104	0.33841463	0.02857143	0
	Test	0.47058824	0.17647059	0.35670732	0.35670732	0.125	0

Table 9 Data generator to train our model

Unrolled index 0:	
Inputs:	[0.86032474 0.79311657 0.79409707 0.8310883 0.90970576 0.79311657]
Output:	[0.86032474 0.90970576 0.90970576 0.09216607 0.90970576 0.90970576]
Unrolled index 1:	
Inputs:	[0.79311657 0.79409707 0.8310883 0.90970576 0.90970576 0.79409707]
Output:	[0.8310883 0.41866928 0.8310883 0.09216607 0.41866928 0.90970576]
…	
…	
Unrolled index 132:	
Inputs:	[0.79409707 0.86032474 0.8310883 0.90970576 0.79409707 0.90970576]
Output:	[0.2070513 0.79409707 0.90970576 0.2070513 0.2070513 0.02807767]
Unrolled index 133:	
Inputs:	[0.8310883 0.79311657 0.90970576 0.8310883 0.8310883 0.79311657]
Output:	[0.2070513 0.41866928 0.02807767 0.90970576 0.2070513 0.79409707]

Table 10 Weights of the input between input and first hidden layers

W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o
Node 1				Node 2				Node 3			
0.378	0.199	0.305	0.506	0.163	0.263	0.979	0.992	0.545	0.576	0.376	0.889
0.059	0.716	0.352	0.521	0.411	0.986	0.836	0.793	0.928	0.174	0.177	0.901
0.399	0.228	0.628	0.243	0.728	0.440	0.519	0.826	0.204	0.838	0.954	0.411
0.663	0.708	0.069	0.910	0.725	0.698	0.181	0.902	0.522	0.695	0.121	0.136
Node 4				Node 5				Node 6			
0.486	0.219	0.191	0.990	0.091	0.648	0.422	0.156	0.515	0.797	0.680	0.883
0.018	0.968	0.928	0.564	0.352	0.405	0.386	0.865	0.940	0.938	0.253	0.580
0.062	0.893	0.098	0.950	0.159	0.136	0.356	0.475	0.951	0.447	0.096	0.827
0.725	0.823	0.400	0.291	0.862	0.312	0.074	0.475	0.303	0.544	0.842	0.279

Table 11 Optimal weights of input between first and second layer

Node 1				Node 2				Node 3				Node 4			
W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o
0.882	0.256	0.604	0.403	0.373	0.999	0.252	0.875	0.350	0.841	0.158	0.912	0.113	0.814	0.287	0.782
0.447	0.412	0.067	0.900	0.223	0.860	0.594	0.913	0.087	0.273	0.174	0.537	0.300	0.895	0.082	0.636
0.338	0.467	0.832	0.209	0.039	0.091	0.547	0.717	0.286	0.918	0.394	0.215	0.603	0.608	0.930	0.460
0.661	0.362	0.657	0.592	0.900	0.763	0.934	0.029	0.542	0.767	0.998	0.794	0.619	0.623	0.899	0.483

Table 12 Optimal weights of input between second and third layer

Node 1				Node 2				Node 3				Node 4			
W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o
0.067	0.724	0.363	0.178	0.709	0.755	0.743	0.843	0.757	0.989	0.547	0.671	0.311	0.930	0.457	0.718
0.422	0.321	0.590	0.597	0.707	0.120	0.699	0.269	0.991	0.030	0.576	0.247	0.238	0.367	0.716	0.740
0.161	0.665	0.109	0.467	0.305	0.992	0.343	0.327	0.053	0.084	0.889	0.092	0.049	0.719	0.816	0.687
0.783	0.388	0.733	0.263	0.130	0.879	0.799	0.906	0.731	0.674	0.420	0.765	0.154	0.443	0.198	0.323

Table 13 Weights of input between third and output layer

Node 1				Node 2				Node 3				Node 4			
W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o	W_a	W_i	W_f	W_o
0.410	0.492	0.598	0.842	0.651	0.777	0.599	0.033	0.456	0.983	0.647	0.569	0.608	0.006	0.192	0.476
0.137	0.681	0.510	0.839	0.741	0.934	0.680	0.817	0.346	0.848	0.984	0.007	0.041	0.557	0.350	0.329
0.801	0.171	0.317	0.079	0.099	0.613	0.227	0.671	0.385	0.218	0.903	0.890	0.053	0.579	0.625	0.936
0.605	0.418	0.610	0.850	0.351	0.303	0.095	0.008	0.152	0.601	0.087	0.990	0.549	0.913	0.731	0.373
0.132	0.830	0.711	0.139	0.092	0.545	0.181	0.717	0.291	0.376	0.722	0.297	0.036	0.084	0.776	0.853
0.456	0.033	0.562	0.703	0.981	0.003	0.254	0.185	0.839	0.431	0.363	0.470	0.792	0.674	0.376	0.420

Table 14 Weight of recurrent connections

U_a	U_i	U_f	U_o
0.390	0.279	0.435	0.622

Table 15 DLSTM-PS and SMAPE results

	PM2.5	PM10	NO_2	CO	O_3	SO_2
DLSTM-RNN	79.29	78.07	73.48	11.60	12.25	14.00
	..	…	…	…	…	…
	67.88	71.18	66.94	7.01	9.37	10.44
SMAPE	13.61	12.25	11.83	4.70	4.52	4.46
	…	…	…	…	…	…
	12.25	11.93	11.61	4.38	4.41	4.22

Compliance with ethical standards

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 authors.

Appendix

Item	Description
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 with diameter less than 2.5 µm
PM10	Particulate matter with diameter less than 10 µm
O_3	Ozone, the unstable triatomic form of oxygen
SO_x	Sulfur oxides
CO	Carbon monoxide
NO_x	Nitrogen oxides
$\odot$	Elementwise or Hadamard product
$\otimes$	Outer product
σ	Sigmoid function
a_t	Input activation
i_t	Input gate
f_t	Forget gate
o_t	Output gate
$State_t$	Internal state
Out_t	Output
W	The weights of the input
U	The weights of recurrent connections
V_i^t:	Velocity of particle i in swarm in dimension j and frequency t
X_i^t	Location of particle i in swarm in dimension j and frequency t
c_1	Acceleration factor related to P_{best}
c_2	Acceleration factor related to g_{best}
r_1^t, r_2^t:	Random number between 0 and 1
t	Number of occurrences specified by type of problem
$G_{\text{best},i}^t$	g_{best} position of swarm
$P_{\text{best},i}^t$	p_{best} position of particle

References

Ali SH (2012) A novel tool (FP-KC) for handle the three main dimensions reduction and association rule mining. In: 2012 6th international conference on sciences of electronics, technologies of information and telecommunications (SETIT), Sousse, IEEE, pp 951–961. https://doi.org/10.1109/SETIT.2012.6482042

Al-Janabi S, Alkaim AF (2019) A nifty collaborative analysis to predicting a novel tool (DRFLLS) for missing values estimation. Soft Comput. https://doi.org/10.1007/s00500-019-03972-x

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), IEEE, pp 37–42. https://doi.org/10.1109/dese.2017.23

Al-Janabi S, Mahdi MA (2019) Evaluation prediction techniques to achievement an optimal biomedical analysis. Int J Grid Util Comput 10(5):512–527

Al-Janabi S, Rawat S, Patel A, Al-Shourbaji I (2015) Design and evaluation of a hybrid system for detection and prediction of faults in electrical transformers. Int J Electr Power Energy Syst 67:324–335. https://doi.org/10.1016/j.ijepes.2014.12.005

Al-Janabi S, Al-Shourbaj I, Salman MA (2018) Assessing the suitability of soft computing approaches for forest fires prediction. Appl Comput Inform 14(2):214–224. https://doi.org/10.1016/j.aci.2017.09.006

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 and systems, vol 81. Springer, Cham. https://doi.org/10.1007/978-3-030-23672-4_10

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

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

Bianchi FM, Maiorino E, Kampffmeyer MC et al (2017) An overview and comparative analysis of recurrent neural networks for short term load forecasting. arXiv:1705.04378

Buyya R, Calheiros RN, Vahid Dastjerdi A et al (2016) Big data principles and paradigms, pp 1–468. https://doi.org/10.1016/C2015-0-04136-3

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

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

Ghoneim OA, Doreswamy, Manjunatha BR (2017) Forecasting of ozone concentration in smart city using deep learning. In: 2017 international conference on advances in computing, communications and informatics (ICACCI 2017), IEEE, pp 1320–1326. https://doi.org/10.1109/ICACCI.2017.8126024

Hu M, Wang H, Wang X et al (2019) Video facial emotion recognition based on local enhanced motion history image and CNN-CTSLSTM networks. J Vis Commun Image Represent 59:176–185. https://doi.org/10.1016/J.JVCIR.2018.12.039

Inácio F, Macharet D, Chaimowicz L (2019) PSO-based strategy for the segregation of heterogeneous robotic swarms. J Comput Sci 31:86–94. https://doi.org/10.1016/J.JOCS.2018.12.008

Li X, Peng L, Hu Y et al (2016) Deep learning architecture for air quality predictions. Environ Sci Pollut Res 23:22408–22417. https://doi.org/10.1007/s11356-016-7812-9

Li X, Peng L, Yao X et al (2017) Long short-term memory neural network for air pollutant concentration predictions: method development and evaluation. J Environ Pollut 231:997–1004. https://doi.org/10.1016/j.envpol.2017.08.114

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

Liu S, Wang Y, Yang X et al (2019) Deep learning in medical ultrasound analysis: a review. Engineering. https://doi.org/10.1016/J.ENG.2018.11.020

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/JCOMPCHEMENG.2019.01.020

Ong BT, Sugiura K, Zettsu K (2015) Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5. Neural Comput Appl 27(6):1553–1566. https://doi.org/10.1007/s00521-015-1955-3

Popoola OAM, Carruthers D, Lad C et al (2018) Use of networks of low-cost air quality sensors to quantify air quality in urban settings. Atmos Environ 194:58–70. https://doi.org/10.1016/j.atmosenv.2018.09.030

Shang Z, Deng T, He J, Duan X (2019) A novel model for hourly PM2.5 concentration prediction based on CART and EELM. J Sci Total Environ 651:3043–3052. https://doi.org/10.1016/j.scitotenv.2018.10.193

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 medicine and 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

Wen C, Liu S, Yao X, Peng L, Li X, Hu Y, Chi T (2019) A novel spatiotemporal convolutional long short-term neural network for air pollution prediction. J Sci Total Environ 654:1091–1099. https://doi.org/10.1016/j.scitotenv.2018.11.086

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

Zhou B-Z, Liu X-F, Cai G-P et al (2019) Motion prediction of an uncontrolled space target. J Adv Space Res 63:496–511. https://doi.org/10.1016/J.ASR.2018.09.025

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.