Measurement: Sensors 24 (2022) 100546

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Air pollution prediction using LSTM deep learning and metaheuristics algorithms

Ghufran Isam Drewil [*], Riyadh Jabbar Al-Bahadili

Computer Engineering Department, University of Technology, Baghdad, Iraq

A R T I C L E I N F O

Keywords:
Deep learning
Long short-term memory (LSTM)
Genetic algorithm (GA)
Time series data
Air pollution

A B S T R A C T

Air pollution is a leading cause of health concerns and climate change, one of humanity's most dangerous problems. This problem has been exacerbated by an overabundance of automobiles, industrial output pollution, transportation fuel consumption, and energy generation. As a result, air pollution forecasting has become vital. As a result of the large amount and variety of data acquired by air pollution monitoring stations, air pollution forecasting has become a popular topic, particularly when applying deep learning models of long short-term memory (LSTM). The ability of these models to learn long-term dependencies in air pollution data sets them apart. However, LSTM models using many other statistical and machine learning approaches may not offer adequate prediction results due to noisy data and improper hyperparameter settings. As a result, to define the pollution levels for a group of contaminants, an ideal representation of the LSTM is required. To address the problem of identifying the best hyperparameters for the LSTM model, In this paper, we propose a model based on the Genetic Algorithm (GA) algorithm as well as the long short-term memory (LSTM) deep learning algorithm. The model aims to find the best hyperparameters for LSTM and the pollution level for the next day using four types of pollutants PM10, PM2.5, CO, and NOX. The proposed model modified by optimization algorithms shows more accurate results with less experience and more speed than machine learning models and LSTM models.

1. Introduction

According to the Urban Population website, people relocating to cities in 2020 was 56.15% [1]. According to the United Nations, cities will house 68% of the world's population by 2050 [2]. This demographic shift would result in many health, transportation, and air quality issues. Air pollution causes many health problems, such as respiratory problems, premature death, and hospitalization for heart and lung diseases [3,4]. Air pollution affects people but dramatically affects plants, as long-term exposure to pollutants causes damage to plant leaves [5]. Most of the contaminants are from stationary sources of the primary air pollutants, which are dust and particles of PM10 with a diameter of fewer than 10 μm and the PM2.5, which are the most dangerous because their diameter is less than PM2.5 microns, which is produced from unburned fuel and process byproducts, as well as sulfur dioxide (SO_2) that is produced From fuel combustion, nitrogen oxides (NO_x), a mixture of oxygen and nitrogen reacting at high temperatures, carbon monoxide (CO) and ozone (O_3) [6].

Air pollution levels must be accurately predicted to work with governments and educate the public about the dangers of pollution. One of the most common ways of describing air pollution data is either a rising or falling trend; seasonality (the variability in time series within a specific time period); cycles (rises and falls that aren't fixed in time); or erratic movement [7].

The deep learning architecture is suitable for solving air pollution prediction problems and nonlinear, cyclical, seasonal, and sequential dependency problems between pollutant data. The LSTM technique is designed to learn long-term dependencies from time-series data, preserving them better than the shallow ANN architecture [7]. A challenge like measuring pollution levels or other variables affected by sequence-dependent behavior is a good fit for the LSTM's internal memory [8]. In other words, the pollution level for each gas is predicted based on previous observations, where a similar pattern of behavior may appear in the future.

One of the essential tasks in developing a deep learning model is determining the LSTM model's hyperparameters. The selection of these hyperparameters affects the model's training process; it is processing in addition to overfitting and thus the accuracy of the final model. In most

* Corresponding author.
E-mail address: ghufranessam95@gmail.com (G.I. Drewil).

https://doi.org/10.1016/j.measen.2022.100546
Received 1 September 2022; Received in revised form 29 September 2022; Accepted 24 October 2022
Available online 29 October 2022

cases, selecting hyperparameters is a complex and time-consuming process to reach the best hyperparameters, which always has an error rate such as choosing the number of layers, number of cells per layer, number of units, batch size, activation type as well as their parameters.

To solve the problem of selecting hyperparameters, especially the window size and number of LSTM units, due to their importance in LSTM inputs and to find a near-ideal model for predicting air pollution, using the Metaheuristics method, which is known for its ability to find near-ideal solutions in large areas.

In this study, one of famous optimization algorithms were applied which is Genetic Algorithm (GA) that will improve the performance of the LSTM model. The LSTM model will be trained with GA to find best window size and LSTM unit and predict the level of air pollution next day for a group of pollutants (PM2.5, PM10, CO, NOx).

2. Related work

Numerous academics worldwide have identified air pollution as a significant environmental concern. Previous research on air quality forecasts can be grouped into physical and empirical models. Aerodynamics, atmospheric physics, and chemistry are used in the physical prediction models to investigate pollution dispersion mechanisms and apply mathematical methodologies to determine spatiotemporal pollutant distributions [9–11].

Empirical models use statistics and prior time-series data to predict future air quality. It is faster and cheaper to make a prediction using a statistical model than a physical model since it avoids complex mechanisms. They also achieve a level of gas concentration prediction accuracy nearly equivalent to physical prediction models [12]. Because of these benefits, statistical prediction models are being used more widely. Le, Van Duc 2020 [13] have proposed the Convolutional Long Short-Term Memory (Conv LSTM) model that converts air pollution data into sequences of images by considering the entire city as one image. This model relies on temporal and spatiotemporal data sets. Several studies suggest using Bi-LSTM [14]; this study found that the air quality index is more significant in winter and more minor in summer by using Bi-directional Long Short_ Term Memory (Bi-LSTM) with several techniques to predict air quality, such as Gated Recurrent Unit (GRU), Convolutional Neural Networks (CNN), Back Propagation Neural Network (BPNN), Long Short-Term Memory (LSTM), and Bi-directional Long Short_ Term Memory (Bi-LSTM) and the best results were obtained using Bi-LSTM. Zhang 2020 [15] In this paper, Empirical Mode Decomposition (EMD) and Bidirectional Long-Short-Term Memory (Bi-LSTM) neural networks are also suggested as semi-supervised model content for EMD-Bi-LSTM. This model may reduce error accumulation in PM2.5–70%. To predict PM2.5 and PM10 constriction for ten days in Seoul, South Korea, Xayasouk 2020 [16] used Deep Autoencoder (DAE) and long short-term memory (LSTM) methods. LSTM gives more accurate results compared to LSTM and DAE using RMAE. In this study, Wu 2019 [17]; has proposed a model for accurately predicting the air quality index. Whereas, Variational mode decomposition (VMD) was used to decompose the AQI series into different frequencies, reconstruct the Sample Entropy SE, and finally predict the AQI level the LSTM algorithm was used. This AQI prediction model showed a high rate of accuracy when compared to individual models.

Abirami 2021 [18]; There are three parts to the suggested model first encoder, second STAA-LSTM, and decoder. The underlying space input is then used to encode all of the linked spatial relationships. Even so, the STAA-LSTM establishes the latent space's relevant temporal relations. The predicted spatiotemporal associations that the decoder deciphered to obtain the expected air quality values are also predicted. DL-Air reduced errors by 30% and improved accuracy by 8%.In this paper [19], the researchers proposed a new model based on LSTM, which is Multi-output and Multi-index of Supervised Learning (MMSL), to predict air quality for the next 9-h mode work by collecting the data from a different station in Beijing that included meteorological data and gaseous pollutant data into the model to realize the concentration prediction of air quality indicators and also the pollution indicator of PM2.5, CO, NO2, O3, SO2.MAE, RMSE, and R2 were used as evaluation metrics. The result showed the efficiency of the MMSL model. Xiao 2020

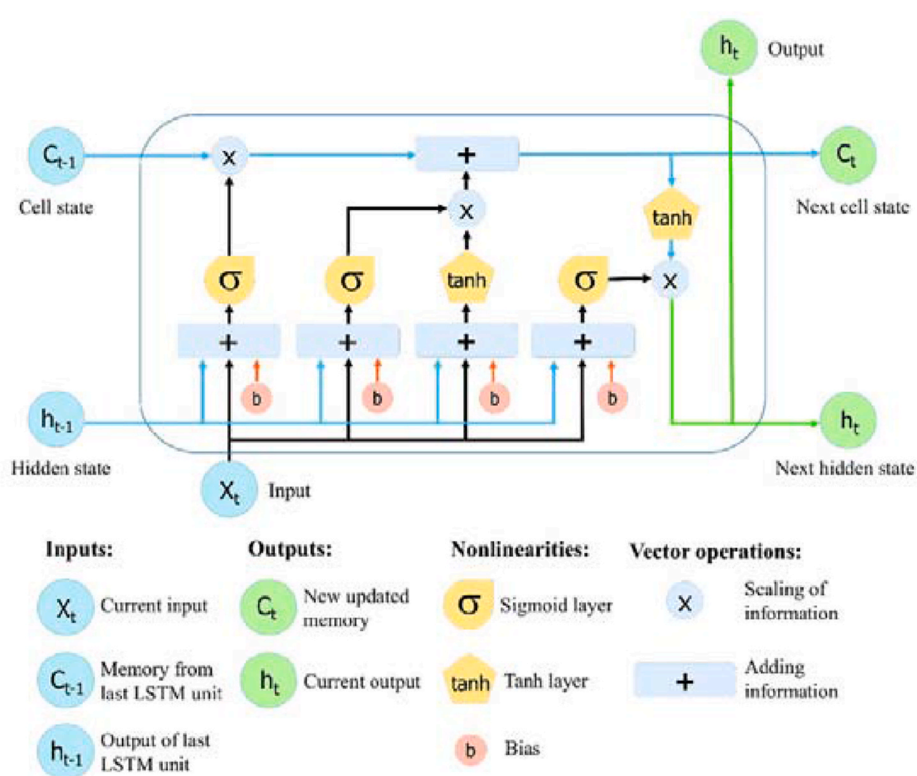

Fig. 1. LSTM memory cell in detail.

[20] introduced a new weighted Long Short-Term Memory Extended model (WLSTME) to predict daily PM2.5 constriction in Beijing–Tianjin–Hebei. Because monitoring stations are not evenly dispersed, a multi-layer perception (MLP) was employed to blend wind speed and direction with PM2.5 data. In the second stage, weighted PM2.5 data is entered into the LSTM, then processed with spatial and temporal dependency and temporal and spatial features extracted. The final step involves combining meteorological data with spatiotemporal information from the primary sites. WLSTME improves accuracy, particularly in the winter.

Several literary studies have used various optimization algorithms; reviewing them, Al-Janabi 2020 [21] The Smart Air Quality Prediction Model (SAQPM) is based on the computation-based particle swarm optimization (PSO) that uses unsupervised learning and the LSTM algorithm to predict air quality. Six pollutants (PM2.5, PM10, SO2, NO2, CO) concentrations are predicted over the next two days. Mao 2020 [22]; The TS-LSTME model (temporal sliding long short-term memory extended) was used to estimate PM2.5 concentrations for one day. Research on air quality monitoring stations' spatiotemporal relationships was largely unsuccessful. This model aims to address the issue. When combined with PM2.5 concentrations per hour and meteorological data, TS-LSTME creates a prediction interval with a multi-layer bidirectional LSTM. When compared to MLR, the evaluation metric R2 produced excellent results. Sun 2020 [23]; to predict nitrogen and sulfur dioxide to provide more accurate information to prevent acid rain using an extreme learning machine with a whale optimization algorithm on eight data sets. This proposed model showed excellent prediction results as the improvement in R2 and RMSE were 897.57% and 50.78%, respectively.

Many studies dealt with this topic in terms of predicting the use of deep learning algorithms. Most studies [12–23] used the LSTM deep learning algorithm. This algorithm proved its efficiency in processing big data and time-series data. One of the defects of this algorithm is that it needs many experiments to get the best hyperparameter to train the neural network, which takes much time. The best way to get the best hyperparameter is to use optimization algorithms and the used one in the is research is PSO algorithms [21]. The PSO and GA have succeeded in various fields. Since they have successfully solved similar problems in other areas, it raises whether the technique can give the best results in terms of hyperparameter for LSTM to forecast air pollution.

3. Background

This section discusses the LSTM model, hyperparameter tuning approaches, GA metaheuristics, and forecasting assessment measures.

3.1. Long Short-Term Memory

LSTM belongs to the category of Recurrent Neural Networks (RNN). RNNs store representations of the most recent entries using feedback, which is why they are called recurrent neural networks because they loop feedback. The idea of LSTM is that it maintains the state of the memory even after a long time due to the presence of the memory cell. The memory state consists of gates that regulate data flow in memory. The memory state is present in all LSTM cells to modify the information values of the previous states based on the importance of the gate units.

The standard LSTM consists of three layers, as shown in Fig. 1. The input layer is the first layer in the model; the second layer is the recurrent layer in LSTM. The output layer is connected to the three gates of the cell (input, output, and forget gates), and finally, the output layer [8].

The equations below show the conventional equations for LSTM

$$f_t = \sigma\left(w_f * [h_t - 1, x_t] + b_f\right) \tag{1}$$

$$i_t = \sigma(\mathrm{w_i}[\mathrm{h_t} - 1, \mathrm{x_t}] + \mathrm{b_i}) \tag{2}$$

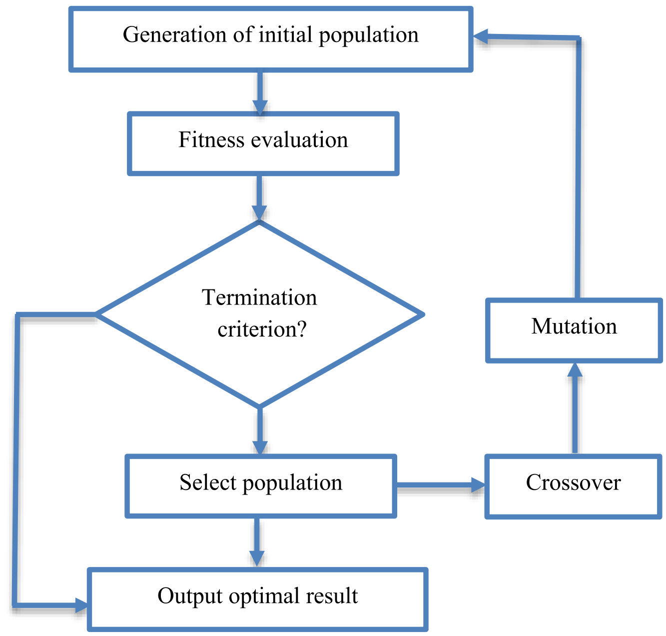

Fig. 2. Basic process of a genetic algorithm (GA) [24,27].

$$\widehat{c}_t = tanh(w_c \cdot [h_t - 1, x_t] + b_c) \tag{3}$$

$$c_t = f_t * C_{t-1} + i_t * \widehat{C}_t \tag{4}$$

$$O_t = \sigma\left(W_0 \cdot \left[h_{t-1,x_t}\right] + b_0\right) \tag{5}$$

$$h_t = O_t * tanh(c_t) \tag{6}$$

Where f_t = forget gate, ϭ = sigmoid, W_f = wight, h_{t-1} = output of previous block, Xt = input vector, b_f = bias. The symbol * signifies elementwise multiplication, C_t = Cell state, h_t = hidden state and O_t = output gate [7].

3.2. Metaheuristic algorithms

Metaheuristics algorithms are popular solutions to complex optimization problems and often provide adequate solutions, especially in light of the restricted resources and short constraints. Optimization algorithms are random algorithms inspired by nature because they monitor natural behavior [7]. This part will explain GA which is a type of optimization algorithms.

3.3. The genetic algorithm

GA is a metaheuristic and stochastic optimization algorithm inspired by Darwinian evolution. GA's work simulates the natural change process to find near-perfect solutions to improve significant search space problems. The primary mechanism of GA work is to start by creating an initial set at random, consisting of a certain number of individuals or "chromosomes." Then, each chromosome's fitness is determined using the fitness function, a concept used to encode the performance of the chromosome numerically. Finally, the mutation operator is applied to the population to introduce diversity and modernity into solutions.

GA can be divided into three stages: population primary production, GA operators (selection, crossover, and mutation), and assessment based on the fitness function. Each chromosome solution is visualized using a series of variable values in the initial population operator. The initial population must accept solutions that preserve each constraint in conditions that improve the problem. For this reason, the selection operator only keeps populations or chromosomes that have shown exceptional

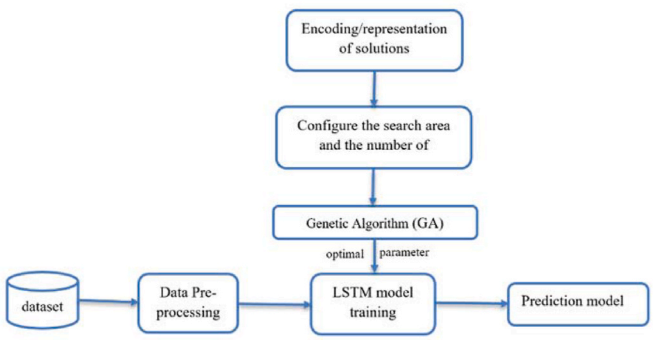

Fig. 3. The proposed system.

performance in the past. One way to create children is by exchanging sections of chromosomes and changing gene combinations in the crossover operator. Switching solution bits on and off randomly is the purpose of this crossover procedure. The chromosomal gene mutation process results in an unexpected change in the gene's corresponding bits. A new chromosome is created through selection, crossover, and mutation, which calculates the model's suitability and checks the termination criteria. These operations are repeated if this chromosome does not meet some termination criteria [7,24–26]. Fig. 2 shows a flowchart of the GA algorithm.

3.4. Evaluation Metrics

Root Mean Squared Error (RMSE) metrics are used to assess a regression model. These measurements show us how accurate the forecasts are and how far off they are from the actual data.

3.5. Root mean squared error

As shown in equation (7) below, the RMSE is a quadratic scoring rule that measures the average magnitude of the error. It is calculated by summing the difference between the forecasted and observed values and then averaging this difference over the sample. Finally, the average square root is computed. Because errors are squared before being averaged, the RMSE gives significant errors a relatively high weight. When large errors are undesirable, the RMSE is especially useful [28].

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(Y_j - \widehat{Y}_j\right)^2} \tag{7}$$

4. Proposed system

In this part, the method of predicting air pollution will be explained, starting with a description of the LSTM deep learning algorithm and GA optimization algorithm and finishing with the techniques and libraries used in the project. we have developed a framework for determining the number of input units for the LSTM model using types of optimization algorithm is GA. In the first step, the solution is represented or encoded. Then the search area and the number of generations are determined. The search component GA processes to find the appropriate number of units for the LSTM model. Fig. 3 show the proposed system.

On the other hand, the data is processed to train the LSTM model. The LSTM model is trained with GA. Then the root means square error (RMSE) was used as a fitness function for each optimization algorithm.

4.1. Data pre-processing

To train the neural network, the historical data set from the Kaggle website was used, which is time-series data for hours for a group of stations in India from 2017 to 2020, containing hourly readings for a group of gases (PM2.5, PM10, CO, NOx, O3, SO2, NO, AQI, AQI-Bucket) Only four of them were used for training and testing (PM2.5, PM10, CO, NOx). As a first step, each station's data was separated into a separate CSV file containing the station name to remove the features that were not used, such as SO2, NO, O3 AQI, and AQI-Bucket using Drop. Then to process the missing values, fill was used to replace the missing values with zero. Using MinMaxScaler property to make the column values in the data set with standard scales without distorting the difference in the range of values and creating the training column between the scale [0,1]. As as final step, the data was divided into 75% training data and 25% test data.

Table 1
LSTM hyperparameter setting.

Hyperparameter	Value
Input time step	24
Input feature dimensions	7
Epoch	200
Batch size	32
Learning rate	0.00001
Optimizer	Adam
Sliding windows	Find the best using GA
LSTM unit	Find the best using GA
Loss function	MSE, RMSE
Dropout	0.2

4.2. Build long short-term memory

As we explained earlier LSTM is a type of RNN; the difference between them is that LSTM contains a long-term memory to hold information through a set of gates (the input gate, the forgetting gate, the output gate) [28]. The hyperparameters must be chosen carefully to achieve good performance with the LSTM network. These parameters affect the neural network's performance, the time cost, and the memory of running deep learning algorithms [29]. Although there are some basic rules for determining hyperparameters, optimization is necessary because these values depend on the data quality. As we can see in Table 1, the number of the Input time step is 24, considering we want to forecast for a day and the number of Input feature dimensions is 7. To find the best LSTM unit for each layer GA will be used, Dropout 0.2 to avoid over-fitting. In addition, it was found that Adam was the best optimizer for this experiment. The epochs and batch size selected for the training model are 200 and 32, respectively.

5. GA-LSTM

GA is a common non-linear optimization algorithm that provides an optimal or near-optimal solution by searching in a complex space. The working principle of GA is based on population research, where a set of candidate solutions for a fitness function is obtained after a series of iterative calculations [30].

There are three basic operations in the genetic algorithm: the crossover operator, the mutation operator, and the selection operator, and these operations are explained in detail in Chapter Two. To make the way GA works with the LSTM clear, we found that the evolution process is iterative, as shown in Fig. 4. The termination criterion is preset with the maximum number of iterations. It can be summarized in the following steps.

1 GA generates the population randomly
2 Every individual in the population is assessed for their fitness evaluation
3 selection operation
4 crossover operation
5 mutation operation

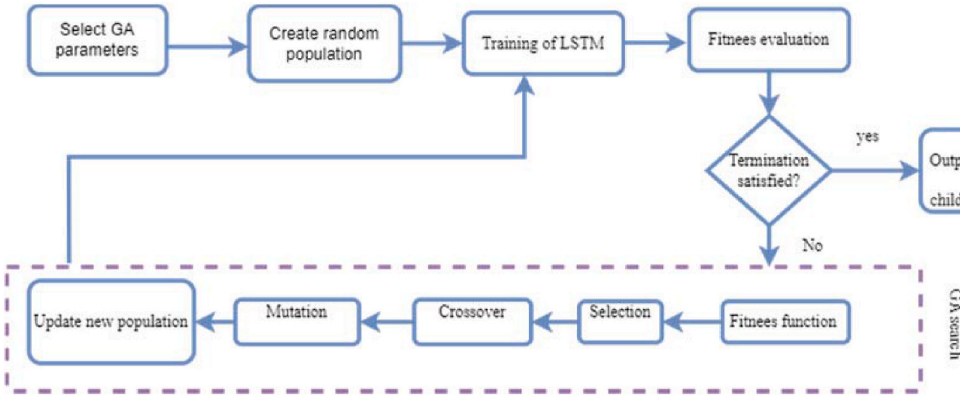

Fig. 4. GA algorithm operation scheme.

Table 2
GA model parameter.

Parameter	Selection
Crossover probability	0.9
Mutation probability	0.6
Selection	selRoulette
Population Size	4
Number of Generations	4
Fitness Function	Root mean square error

6 trains the LSTM
7 Stop GA if the completion criteria are met. If it does not meet the requirements, repeat the process from step 2

The proposed model attempts to improve the prediction error in the LSTM algorithm by designing a hybrid GA-LSTM model to find the number of LSTM units in each layer and the appropriate window size. The first step is processing the data, defining the intervals, converting the data into a supervised model, and splitting the training and test data.

The next step is to train the GA-LSTM model with the initial window

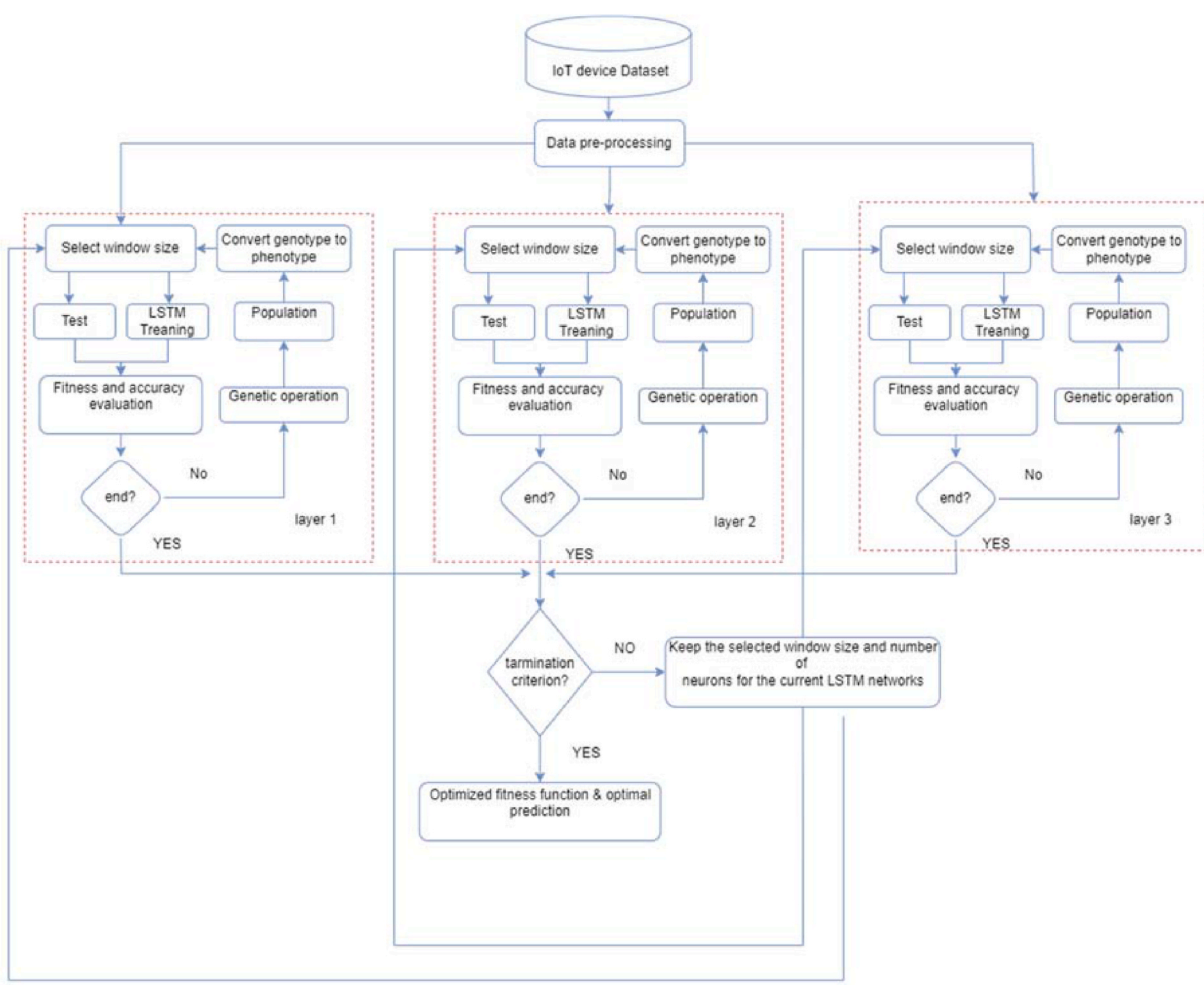

Fig. 5. The GA-LSTM optimization architecture with three layers.

Table 3
Suggests hyperparameter.

Hyperparameter	Best Value
window size	37
No of the LSTM units	9

Table 4
Evaluation results of the benchmarks and the GA.

Metrics	GA-LSTM	Bi-LSTM [15]	CLSTM [6]	WLSTEM [20]
RMSE	9.5820	22.58	13.97	40.67
MAE	19.164	16.67	3.54	26.10

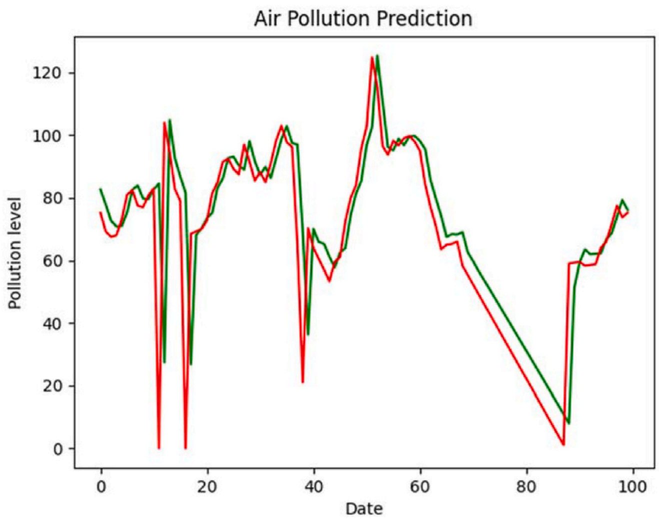

Fig. 6. Forecast GA-LSTM, original data VS. prediction data.

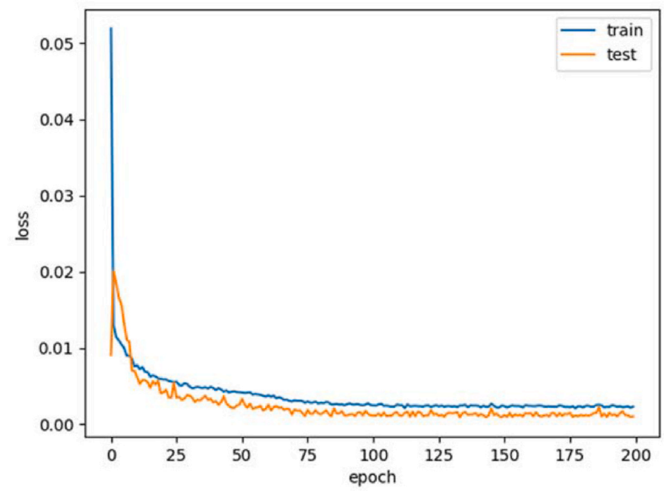

Fig. 7. Loss function for the training and testing of LSTM.

size and the number of LSTM units in the first layer. After that, the model is tested with the specified window size and the number of LSTM units, and the accuracy is calculated using the squared mean and optimized adam. The hyperparameters used in LSTM are shown in Table 1. The number of epochs for all 200 models when one period is a complete pass for all training data.

The window size and the number of units are dynamic, not static values. For that, we have used Genetic Algorithm to get the best window size and number of units. Window size and the number of units of LSTM were evaluated synchronically in training mode. In other words, if window size and the number of units do not meet the perfect Mean Square Error, the genetic algorithm will evaluate another window size and number of units. Thus, LSTM will run with superior accuracy. Table 2 show the GA model parameter.

The evolution rule in GA searches for the best solutions using the evolutionary concepts, crossover, mutation, and selection to generate window size and the number of units in each hidden layer. Fig. 5 illustrates how LSTM GA works using 3 LSTM layers. This model generates new chromosomes with window size and the number of LSTM units to strengthen the search dynamism and improve prediction accuracy.

6. Rustle

The LSTM was trained with GA optimization algorithms as we explained in section two. The LSTM model was created using the Python programming language and the Keras application programming interface (API), an API geared at humans and runs atop the TensorFlow backend.

Suggested parameters to be improved are suggested from the GA model have been described in Table 3 which suggests window size 37, and the number of units for LSTM is 9.

Next, using these parameters suggested by GA, the LSTM model was trained. And as we explained previously, the data was divided into 75 training data and 25 test data. You notice in Fig. 6 the difference between the actual and training data. Figure A shows GA-LSTM. In the figure below, the green color line represents the actual data, and the red line represents the test data. The X-axis represents the time periods, and the Y-axis represents the pollution level.

Finally, the performance of the GA-LSTM model was evaluated using the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) Evaluation Metrics and compared with the models that use manual criteria, as shown in Table 4. Comparing the proposed model with the research papers models that we reviewed in section 1, we note that the GA optimization algorithm has significantly improved the prediction of LSTM.

The multi-sequence input proposed by GA indicates that the air pollution data set has a multi-scale characteristic. It can show distinct patterns of the pollutant sequence in different time scales by using GA.

When the model performance on both the training set and the testing set is good, and when the model does not overfit, the model's fit is regarded as acceptable. Fig. 7 shows the learning curve, where the decay of the loss function with epoch is noticed. This figure shows if the model has been incorrectly prepared for the training and test data set. Moreover, it could be noted in this plot that the loss decreases very quickly due to the size of the training data, and it will settle in the same snapshot. This implies that the proposed model was successful in predicting air pollution.

7. Conclusions

To achieve the paper objective; A new approach is proposed to use LSTM to predict air quality for the next day. One of the problems of the LSTM algorithm is the selection of parameters, window size, and the number of units in LSTM, where it was found that metaheuristics GA was a successful solution to finding the best parameters. Since the LSTM input sequences are of fixed length, using GA leads to a more flexible performance which is necessary when predicting pollution levels. It thus leads to more learning for long-term predictions of air pollution patterns. Also, the results of the comparison with different models of LSTM showed that the results of air pollution prediction based on the metaheuristic principle give better prediction results than that which works to determine the parameters manually.

CRediT authorship contribution statement

Ghufran Isam Drewil: Conceptualization, Methodology, Software, Validation, Formal analysis, Resources, Writing – original draft, Writing – review & editing, Visualization. **Riyadh Jabbar Al-Bahadili:** Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ghufran reports financial support was provided by University of Technology. Dr. Riyadh Jabbar Albahadili reports a relationship with University of Technology that includes: employment.

Data availability

The authors do not have permission to share data.

References

[1] United Nations Population Division, The world bank, Urban population (% of total population). https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS?end=2020&name_desc=false&start=1960&view=chart, 2020 accessed Sep. 25, 2021.

[2] UN, "United Nations. https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html, 2018 accessed Sep. 25, 2021.

[3] I. Manisalidis, E. Stavropoulou, A. Stavropoulos, E. Bezirtzoglou, Environmental and health impacts of air pollution: a review, Front. Public Health 8 (2020), https://doi.org/10.3389/fpubh.2020.00014.

[4] A.M. Haleem, et al., Air quality assessment of some selected Hospitals within Baghdad City, Eng. Technol. J. 37 (1) (2019) 59–63, https://doi.org/10.30684/etj.37.1C.9.

[5] A.-H.M.J. Al-Obaidy, I.M. Jasim, A.-R.A. Al-Kubaisi, A.H.M.J. Al-Obaidy, I. M. Jasim, A.R.A. Al, Air pollution effects in some plant leave morphological and anatomical characteristics within Baghdad City, Eng. Technol. J. 37 (1) (2019) 84–89, https://doi.org/10.30684/etj.37.1C.13.

[6] E. Sharma, R.C. Deo, R. Prasad, A.V. Parisi, N. Raj, Deep air quality forecasts: suspended particulate matter modeling with convolutional neural and long short-term memory networks, IEEE Access 8 (2020) 209503–209516, https://doi.org/10.1109/ACCESS.2020.3039002.

[7] S. Bouktif, A. Fiaz, A. Ouni, M.A. Serhani, Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting, Energies 13 (2) (2020) 1–23, https://doi.org/10.3390/en13020391.

[8] Y.A. Ayturan, Z.C. Ayturan, H.O. Altun, Air pollution modelling with deep learning: a review, Int. J. Environ. Pollut. Environ. Model. 1 (3) (2018) 58–62.

[9] Analysis of Changes in Air Pollution Quality and Impact of COVID-19.Pdf.".

[10] H. Li, et al., Analyzing the impact of heating emissions on air quality index based on principal component regression, J. Clean. Prod. (2017), https://doi.org/10.1016/j.jclepro.2017.10.106.

[11] S. Wang, Y. Li, Evidence on the Impact of Winter Heating Policy on Air Pollution and its Dynamic Changes in North China, 2019.

[12] J. Engel-cox, N. Thi, K. Oanh, A. Van Donkelaar, R. V Martin, E. Zell, Toward the next generation of air quality monitoring : particulate Matter, Atmos. Environ. 80 (2013) 584–590, https://doi.org/10.1016/j.atmosenv.2013.08.016.

[13] V.D. Le, T.C. Bui, S.K. Cha, Spatiotemporal deep learning model for citywide air pollution interpolation and prediction, Proc. - 2020 IEEE Int. Conf. Big Data Smart Comput. BigComp 2020 (2020) 55–62, https://doi.org/10.1109/BigComp48618.2020.00-99.

[14] C. Zhan, S. Li, J. Li, Y. Guo, Q. Wen, W.S. Wen, Prediction of air quality in major Cities of China by deep learning, Proc. - 2020 16th Int. Conf. Comput. Intell. Secur. CIS 2020 (July) (2020) 68–72, https://doi.org/10.1109/CIS52066.2020.00023.

[15] L. Zhang, P. Liu, L. Zhao, G. Wang, W. Zhang, J. Liu, Air quality predictions with a semi-supervised bidirectional LSTM neural network, Atmos. Pollut. Res. (August) (2020) 1–12, https://doi.org/10.1016/j.apr.2020.09.003.

[16] T. Xayasouk, H.M. Lee, G. Lee, Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models, Sustain. Times 12 (6) (2020), https://doi.org/10.3390/su12062570.

[17] Q. Wu, H. Lin, Daily urban air quality index forecasting based on variational mode decomposition , sample entropy and LSTM neural network, Sustain. Cities Soc. 50 (March) (2019), 101657, https://doi.org/10.1016/j.scs.2019.101657.

[18] S. Abirami, P. Chitra, Regional air quality forecasting using spatiotemporal deep learning, J. Clean. Prod. 283 (2021), 125341, https://doi.org/10.1016/j.jclepro.2020.125341.

[19] D. Seng, Q. Zhang, X. Zhang, G. Chen, X. Chen, Spatiotemporal prediction of air quality based on LSTM neural network, Alex. Eng. J. 60 (2) (2021) 2021–2032, https://doi.org/10.1016/j.aej.2020.12.009.

[20] F. Xiao, M. Yang, H. Fan, G. Fan, M.A.A. Al-qaness, An improved deep learning model for predicting daily PM2.5 concentration, Sci. Rep. 10 (1) (2020) 1–11, https://doi.org/10.1038/s41598-020-77757-w.

[21] S. Al-Janabi, M. Mohammad, A. Al-Sultan, A new method for prediction of air pollution based on intelligent computation, Soft Comput. 24 (1) (2020) 661–680, https://doi.org/10.1007/s00500-019-04495-1.

[22] W. Mao, W. Wang, L. Jiao, S. Zhao, A. Liu, Modeling air quality prediction using a deep learning approach: method optimization and evaluation, Sustain. Cities Soc. (2020), 102567, https://doi.org/10.1016/j.scs.2020.102567.

[23] W. Sun, C. Huang, A hybrid air pollutant concentration prediction model combining secondary decomposition and sequence reconstruction, Environ. Pollut. 266 (2020), 115216, https://doi.org/10.1016/j.envpol.2020.115216.

[24] H. Chung, K.S. Shin, Genetic algorithm-optimized long short-term memory network for stock market prediction, Sustain. Times 10 (10) (2018), https://doi.org/10.3390/su10103765.

[25] M. Abdel-Basset, L. Abdel-Fatah, A.K. Sangaiah, Metaheuristic Algorithms: A Comprehensive Review, Elsevier Inc., 2018, https://doi.org/10.1016/B978-0-12-813314-9.00010-4.

[26] H. Yang, M. Hasanipanah, M.M. Tahir, D.T. Bui, Intelligent prediction of Blasting-induced ground vibration using ANFIS optimized by GA and PSO, Nat. Resour. Res. 29 (2) (2020) 739–750, https://doi.org/10.1007/s11053-019-09515-3.

[27] X.H. Shi, Y.H. Lu, C.G. Zhou, H.P. Lee, W.Z. Lin, Y.C. Liang, Hybrid evolutionary algorithms based on PSO and GA, 2003 Congr. Evol. Comput. CEC 2003 - Proc. 4 (2003) 2393–2399, https://doi.org/10.1109/CEC.2003.1299387.

[28] R. Bhuiyan, Examination of Air Pollutant Concentrations in Smart City Helsinki Using Data Exploration and Deep Learning Methods, 2021.

[29] M. Husein, I. Chung, Day-ahead solar irradiance forecasting for microgrids using a long short-term memory recurrent neural network, A Deep Learning Approach (2019), https://doi.org/10.3390/en12101856.

[30] A. Almalaq, J.J. Zhang, Evolutionary Deep Learning Based Energy Consumption Prediction for Buildings, IEEE Access, December 2018, p. 1, https://doi.org/10.1109/ACCESS.2018.2887023, 2019.