International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I2P234
Volume 71 Issue 2, 323-333, February 2023


*Original Article*

# Optimizing Long Short-Term Memory Network Parameter with Artificial Bee Colony Algorithm for PM2.5 Prediction

Thanpitcha Atiwanwong[1*], Adirek Jantakun[1], Adisak Sangsongfa[2]

[1]*Department of Electronics and Telecommunication Engineering, Faculty of Engineering, the Rajamangala University of Technology Isan, Khon Kaen Campus, Khon Kaen, Thailand, 40000, Thailand.*
[2]*Department of Computer Science, Rajabhat Mu-Ban Chombueng University, Ratchaburi, 70150, Thailand.*

[1*]*Corresponding Author : thanpitcha.at@rmuti.ac.th*

Received: 19 December 2022 Revised: 13 February 2023 Accepted: 17 February 2023 Published: 25 February 2023

***Abstract*** - *The problem caused by the PM2.5 value exceeding the standard causes an impact on the population, whether in terms of health, economics, and social aspects, not only in Thailand but all over the world. To solve the problem of PM2.5 occurrence well, it is necessary to be able to predict PM2.5 occurrence effectively. Therefore, this research presents the prediction of PM2.5 occurrence using the Long Short Term Memory neural network. Parameter optimization with the Artificial Bee Colony algorithm, the experimental results obtained an average accuracy of 98%, an average error (MSE) of 0.00267, and an overall parameter value of 6,476 params. The experimental results were compared with the Long Short Term Memory neural network that used experts to determine the parameters. The results showed that the average accuracy was 96%, the average error (MSE) was 0.00651, and the total number of parameters was 21,025 params.*

***Keywords*** - *Artificial Neural Network, Long Short Term Memory, Artificial Bee Colony Algorithm, PM2.5*

## 1. Introduction

What is the importance and problem of PM 2.5? PM (Particulate Matter) is the "Final dust", that also call as PM2.5, established as a standard by The United State Environment Protection Agency, which is particulate matter in the air with a diameter of 2.5 microns and it is harmful to health [1]. A study of air pollution by the Institute for Health and Evaluation, University of Washington [2], found that air pollution is causing a contributor to many diseases. Examples of diseases are ischemic heart disease, cerebrovascular disease, and lung cancer, including acute infectious diseases and lower respiratory tract ozone gas, which is a lung irritant.

From the importance of PM2.5 problems that affect human health and family members, therefore, some researches help in the management of PM2.5, whether it is forecasting or predicting the occurrence of PM2.5. These researches have the same aim: to prevent dust generation or to reduce dust generation in the future. Therefore, these relevant research examples can be illustrated as follows. Yanlin Qi [3] presented the research that used a hybrid model based on deep learning methods that integrate a Convolutional neural network and Long Short-Term Memory networks (GC-LSTM) to model and forecast the spatiotemporal variation of PM2.5 concentrations. The researcher made random division on spatiotemporal blocks (each block consists of a T graph in time ordering) rather than original observations. The researcher proposed that this methodology can be used for concentration forecasting of different air pollutants in the future. Dewen Seng et al. [4] presented their research, a model of spatiotemporal prediction hourly in Beijing, China. The researcher collected data from 35 air quality monitoring stations from January 1, 2016, to December 31, 2017, for this study. The researcher used long short-term memory (LSTM), and multi-output and multi-index supervised learning (MMSL), from the dataset of 35 stations to validate the model's performance compared with other models and two advanced models.

Moreover, research results show that in overall performance, the developed model was MAE and RMSE better than baseline models such as SVM, ARMA, and MMSL.

Furthermore, some researchers researched creating a tool to collect PM2.5 that is inexpensive and high-performance. Thanpitcha Atiwanwong and Saweth Hongprasit [5] their research presented a low-power real-time pollution monitoring system using ESP LoRa to collect PM2.5 in Khon Kaen, Thailand. The research result shows that the tool has a performance standard equal tool of the Pollution Control Department and can report PM2.5 values in real-time on a webpage. The observations from the research above are particularly important due to a matter of importance to human health, and all studies aimed at predicting the incidence of PM2.5 particulate matter to determine ways to prevent the harmful effects of PM2.5 on human health. Syed Ahsin Ali Shah et al. [6] presented their research which on the forecasting of particulate matter. A hybrid model for forecasting particulate matter concentrations based on multiscale characterizations and machine learning techniques using machine multi-method techniques. Learning combined with multiscale characterization, for example, random forest, support vector regressor,[17] k-nearest neighbours, feed-forward neural network, and AdaBoost. The algorithm used by the researchers used the empirical mode decomposition. The developed algorithm, compared with the standard ML method, found that the RMSE value of PM2.5 was 4.81 and MAE was 3.02, which was effective in predicting the occurrence of PM2.5 particulate matter to remind people to protect themselves.

However, another important aspect of the technique used for predicting PM2.5 by the machine learning method is the optimization of the parameters, which results in good output values. Therefore, the most widely used machine learning parameters for predicting PM2.5 data are for experts to determine those parameters and experiment until the best output is obtained. However, the expert configuration is difficult to configure parameters because specialists cannot be proficient in every machine-learning method. Therefore, to optimize the parameters in machine learning in the age of large data usage and very complex, it is necessary to find an algorithm that can help in which heuristic algorithm is the best choice. Because searching for the answer of the heuristic algorithm will not look at all the data until it selects the most suitable answer for the search, the disadvantage is that the answer may not be the best answer [7]. Here are some well-known examples of the intelligent algorithm used in heuristic algorithms to solve complex and large-scale problems that some algorithms use to mimic animal behaviour like prey, which can be sampled as follows Swarm Intelligence, Tabu Search, Genetic Algorithms, Artificial Neural Networks and Support Vector Machines [8]. There are some examples of research related to the optimization of parameter value to obtain the best output values using heuristic algorithms for PM2.5 prediction: Feng Jiang et al. [9] presented an interesting study on the use of the Sparrow Search Algorithm (FOOSSA) to optimize the autoregressive recurrent network (DeepAR) to predict the value of PM2.5. The results obtained from the experiment revealed that the proposed FOSSA–DeepAR learning model could achieve more efficient and accurate predictions in both interval and point prediction, where FOSSA–DeepAR reduces the RMSE, MAE, and MAPE by an average of 2.834%, 6.243%, and 22.55%, respectively. Jie Heng and Min Li [10] presented a research paper titled prediction of PM2.5 concentration based on d Back Propagation Neural Network (BP) optimized by bee colony algorithm. The results showed that when a bee colony was used to determine the optimal BP, the predictions were MSE = 333.43%, RMSE = 18.20% and MAE = 14.32%, respectively. Sang Won Choi and Brian H. S. Kim [11] presented research on PM2.5 prediction in which the researchers used long short-term memory (LSTM) type recurrent neural network (RNN) deep learning and principal components analysis (PCA) to determine the optimal value. This research used daily data for five years to predict PM2.5 concentrations in eight Korean cities. The researcher proposed a PCA-applied model to solve this problem through performance comparison with a non-PCA model, showing that PCA applications produce better results in deep learning time series prediction. From the experimental results, it was found that RMSE = 16.6% and MAE = 33.3%, which were the values obtained from using PCA to find the appropriate value for LSTM. WANG Qiang et al. [12] present research findings related to PM2.5 prediction using the BP-Neural Network Model, which optimizes parameters with Artificial Bee Colony for high accuracy output values, where RMSE = 18.39564 was measured and without Artificial Bee Colony, it was RMSE = 19.31662. From the experiments, it was concluded that using Artificial Bee Colony for optimization was better than the determination by experts.

From the research presented above, it was found that the research related to the PM2.5 prediction was based on finding the optimal value of the parameters of the neural network using the heuristic algorithm better output than using expert parameterization. Furthermore, it was found that the neural network system became more complex and required good output values such as LSTM. From research, it was found that Xin-She Yang commented, “D. Karaboga first developed the ABC optimization algorithm in 2005. Since then, Karaboga and Basturk and their colleagues have systematically studied the performance of the ABC algorithm concerning unstrained optimization problems and its extension [13].” Bharat comments in his book LPWAN Technologies for IoT and M2M Applications, “ABC has been widely used in several applications in many different fields such as training neural networks, signal processing applications, and machine learning community [14].”

This PM 2.5 problem began to become severe in Thailand around the year 2018, that is, about 3 years ago, continuously every year, starting when entering winter around December and will subside when the weather heats up for about a month in April [15]. Because solving the problem of PM2.5 is very important for researchers. To solve the problem of PM2.5, it is good to try to reduce the occurrence of PM2.5 as much as possible, and to reduce the occurrence of it is good to try to predict when PM2.5 will occur. What will be the trend? Therefore, in this research, the prediction of the incidence of PM2.5 in Thailand, especially in Bangkok, was presented to utilize this research to solve the occurrence of PM2.5 problems. Therefore, this research needs to use Artificial Bee Colony Algorithm to adjust the LSTM parameters for predicting PM2.5 to obtain the best output value from the maximum accuracy and minimize MSE value.

## 2. Related Work

### *2.1. Long Short-Term Memory Network: LSTM*

Because Recurrent Neural Network (RNN) is very popular for use with text data, especially for Language Modeling and Machine Translation applications. The actual work found problems arising from the use of too many sequences of data in the analysis. This causes the resulting gradient value to change so that the change cannot be observed and cannot be used for analysis. The problem above is the problem of Gradient Vanishing. Subsequently, the RNN was modified to solve the problem of Gradient Vanishing by adding additional functionality, the Long Short-Term Memory algorithm. LSTM [16] is an algorithm developed from RNN to solve the problem of Gradient Vanishing by designing a new cell function to store the state of the computation. The LSTM cell has a sub-computing unit. It is called a Gate, consisting of an Input Gate, Forget Gate, Memory Cell State Gate, Output Gate, etc. As shown in Figure 1. While the Input Gate is a sub-unit that defines the data to be analyzed in the cell, which receives the input and writes the value into each cell as shown in Equation 1 as follows: [28]

$$i_t = \sigma\left(W_{x^i}x_t + W_{h^i}h_{t-1} + W_{c^i}C_{t-1} + b_i\right) \quad (1)$$

Where $i_t$ is the result of the input gate.

$\sigma$ is the Sigmoid function.

$W_{x^i}$ is the weight value for calculating the input in the input gate.

$x_t$ is the input value to be calculated.

$W_{h^i}$ is the weight for calculating the hidden state in the input gate.

$h_{t-1}$ is the hidden state value obtained from the previous time unit calculation.

$W_{c^i}$ is the weight for calculating the memory cell state in the input gate.

$C_{t-1}$ is the Memory cell state value obtained from the calculation in the previous unit of time.

$b_i$ is the bias value used in the calculation of the input gate.

Forget gate is a unit used to define the data to be analyzed in the cell. Determining whether the data should be saved or forgotten can be determined from Equation 2 as follows:

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \quad (2)$$

Where $f_t$ is the result of the forget gate.

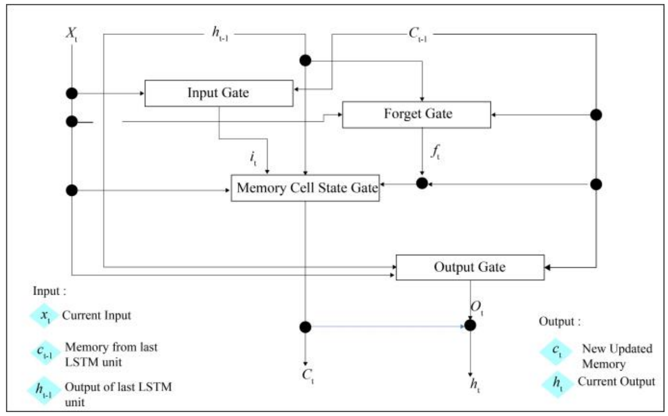



**Fig. 1 The structure of Long Short-Term Memory Neural Network**

$\sigma$ is the Sigmoid function.

$W_{xf}$ is the weight value for calculating the input in the forget gate.

$x_t$ is the input value to be calculated.

$W_{hf}$ is the weight for calculating the hidden state in the forget gate.

$h_{t-1}$ is the hidden state value obtained from the previous time unit calculation.

$W_{cf}$ is the weight for calculating the memory cell state in the forget gate.

$c_{t-1}$ is the memory cell state value obtained from the calculation in the previous unit of time.

$b_f$ is the bias value used in the calculation of the forget gate.

Memory Cell State Gate is a sub-unit to define the data imported to analyze in the cell and calculate the state value for use in the next calculation with the equation as shown in Equation 3 as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot tanh\left(W_{x^c}X_t + W_{h^c}h_{t-1} + b_c\right) \quad (3)$$

Where $C_t$ is the memory cell state value in a time unit.

$f_t$ is the result obtained from forget gate.

$C_{t-1}$ is the Memory cell state value from the previous unit of time.

$i_t$ is the result obtained from the input gate.

$tanh$ is the Hyperbolic tangent function.

$W_{x^c}$ is the weight value for calculating the Input value from the memory cell state gate.

$X_t$ is the input value to be calculated.

$W_{h^c}$ is the weight value for calculating the hidden state in the memory cell state gate.

$h_{t-1}$ is the hidden state value obtained from the previous time unit calculation.

$b_c$ is the bias value used in the calculation of the forget gate.

Output Gate is a sub-unit for calculating the Output of a Cell. The results obtained from this cell have two outputs: Output and Hidden State for use in the next calculation. Equations 4 and 5 are as follows:

$$\boldsymbol{O_t = \sigma(W_{x^o}x_t + W_{h^o}h_{t-1} + W_{c^o}C_{t-1} + b_o)} \quad (4)$$

$$\boldsymbol{h_t = o_t \cdot tanh(c_t)} \quad (5)$$

Where $\boldsymbol{O_t}$ is the result of the output gate.
$\boldsymbol{\sigma}$ is the Sigmoid function.
$\boldsymbol{W_{x^o}}$ is the weight value for calculating the input in the output gate.
$\boldsymbol{x_t}$ is the input value to be calculated.
$\boldsymbol{W_{h^o}}$ is the weight for calculating the hidden state in the output gate.
$\boldsymbol{h_{t-1}}$ is the hidden state value obtained from the previous time unit calculation.
$\boldsymbol{W_{c^o}}$ is the weight for calculating the memory cell state in the output gate.
$\boldsymbol{C_{t-1}}$ is the memory cell state value obtained from the calculation in the previous unit of time.
$\boldsymbol{b_o}$ is the bias value used in the calculation of the output gate.
$\boldsymbol{h_t}$ is the hidden state value to be calculated.

### *2.2. Artificial Bee Colony Algorithm: ABC Algorithm*

ABC algorithm is inspired by the ingenious behavior of bee's swarms, a popular swarm-based technique to solve optimization problems in the real world since 2005. [18] Many researchers are interested in ABC to use algorithms for solving problems, test process optimization, and automation because the ABC algorithm is simple and efficient. In particular, the ABC algorithm is used in software testing to increase efficiency with other techniques for maximum efficiency. There is extensive literature on hybrid approaches from the ABC algorithm and other techniques. [19]

ABC algorithm is an algorithm developed by Karaboga [20] in 2005. The ABC algorithm simulates the bee's foraging behavior to apply it to find the best solution or method for a problem. This algorithm is a metaheuristic method in which bees are divided into 3 types: employee bees, onlooker bees and scout bees. Each type of bee has different responsibilities, as follows. The workflow starts with the employer bees tasked with searching for food sources, estimating the amount of food in the discovered food sources, and then returning to report the discovered food sources to onlooker bees. Onlooker Bees will then select the most suitable food source, where a food source with a high content compared to other nearby food sources is highly likely to be selected, and scout bees will continue to explore the new food source. When the amount of feed comparable to the hyper parameters that employee bees find does not change for the better, the ABC algorithm automatically adjusts the hyperparameters. A set of hyperparameters is like a bee's food source, with each set of hyperparameters having a fitness value comparable to food intake; onlooker bees select a high-suited set of hyperparameters as if they were selecting a high food source which is called the roulette wheel selection. The procedure for working the ABC algorithm can be written as a pseudo-code as follows in Figure 2 [21].

## 3. Methodology

In this experimental algorithm, the researcher designed the frame of work to illustrate the overall research approach from Figure 3. Considering the frame of work, it was found that this research started by collecting the PM2.5 data first. After obtaining the data, it led to the development of predicting the occurrence of PM2.5 with the LSTM neural network. Predicted using LSTM and then considered all the parameters used to select the optimal value using the Artificial Bee Colony algorithm to obtain the next best output value as detailed in each. The steps are as follows.

```
Initialization:
Initialise the initial population and evaluate the fitness;
Calculate the initial fitness value, f(Sol); Set best solution, Solbest← Sol;
Set maimum number of iterations, Numofite;
Set the population size;
Where population size = # of onlooker bees = # of employee bees
Iteration <- 0;
Improvement:
Do while (iteration<NumOfIter)
  for i=1: # of employee bees
     Select a random solution and apply
     random neighborhood structure
end for
for i=1: # of onlooker bees
     Sol * <- select solution from population
     based on roulette wheel
     Apply a random neighborhood structure
    on Sol*:
 if (f(Sol)<f(Sol*))
     Sol* <-Sol;
  end if
  end for
   Solbest <- best solution found so far ;
   Scout bees determine the abandoned food
   source and replace it with the new food source.
   iteration++
End do
```

**Fig. 2 The pseudocode of the Artificial Bee Colony Algorithm**

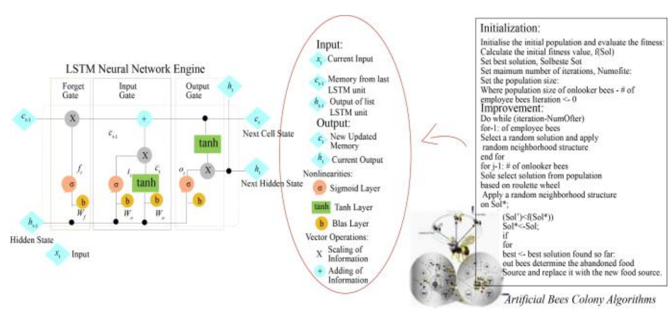



**Fig. 3 Optimization Long Short-Term Memory Neural Network Parameter with Artificial Bee Colony Algorithms for PM2.5 Prediction Frame Work**

### *3.1. Data Preparation*

To prepare data for use in this research, the researcher collected data on the value of PM2.5 from the Pollution Control Department of 12 stations, which are Bangkok's weather stations. However, the researcher chose 7 stations that are expected to be dense, dusty locations, and these 7 stations cover densely populated areas and traffic in Bangkok. All the stations that collected the data were using historical data of PM2.5 from August. 2019 – December 2021, respectively. Once the data is obtained, the researcher uses the data to clean it before it is used for processing, such as removing the period of no data, changing the data type to the same type, removing the column that is not needed for processing, etc. The details of the stations used by the researcher for predicting Bangkok's PM2.5 values are listed below.

1) Chulalongkorn Hospital Station (ST01)
2) Khlong Chan Community Housing Station (ST02)
3) Public Relations Department Station (ST03)
4) Thonburi Substation (ST04)
5) Nonsi Witthaya School Station (ST05)
6) Bangna Meteorological Department Station (ST06)
7) Station on Kanchanaphisek Road, Bang Khun Thian District (ST07)

Where all seven stations in the above use a total of 20,000 records. However, Figures 4 and 5, with the data cleaned, show the source and the data in the form of a graph plot.

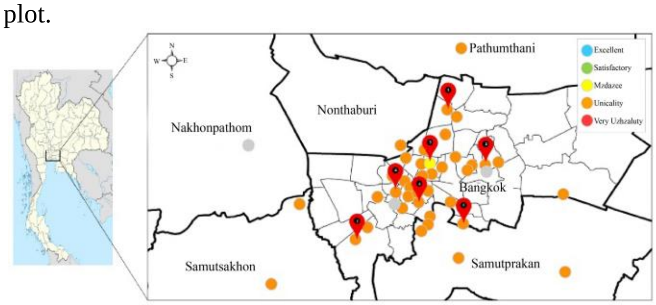



**Fig. 4 Map of 7 representative weather stations in Bangkok**

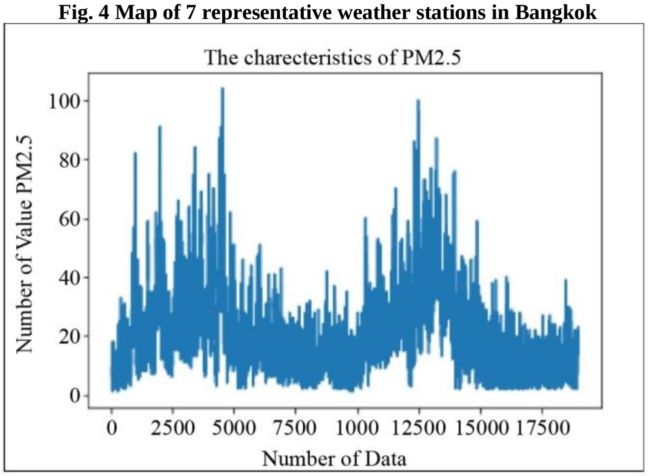



**Fig. 5 The figure shows an example plot of PM2.5 data.**

### *3.2. Measuring the efficiency of the Neural Network*

In this research, the researcher chose to measure the efficiency of the neural network from two types: accuracy and mean squared error (MSE), as detailed below.

• Prediction accuracy is the ratio of the number of prediction accuracy to the total number of predictions. The equation is as follows:

$$\textit{prediction accuracy} = \frac{\textit{correct prediction}}{\textit{total prediction}} x\ 100\% \quad (6)$$

Mean Squared Error (MSE) measures error by squaring the error value and bringing it to its mean. To measure the accuracy of this method, the smaller the value obtained, the more accurate the model will be. The equation is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (7)$$

Where, $y_i$ is the actual value used in the prediction test.
$\hat{y}_i$ is the value that the model predicted out of the test.
$n$ is all used in the test.

### *3.3. Using the Artificial Bee Colony Algorithm (ABC Algorithm) procedure to adjust the parameters of the LSTM*

In this section, the researcher divided the work into two parts: 1) Configure the parameters of LSTM by using expert values to determine the accuracy and MSE values. 2) Use ABC's algorithm to optimize LSTM parameters to determine the accuracy and MSE values.

#### *3.3.1. Configure the parameters of LSTM by using expert values.*

Configure the parameters of the LSTM containing inputshape=120 nodes, dense=32, and in the hidden layer; both layers contain dense=32. The output layer has the value =1. At the same time, the learning rate controls each learning step as lr = 0.00001, optimizer = Adam, loss = MSE, and accuracy, respectively. Use the total number of learning cycles for 150 epochs. Therefore, the parameters configuration of LSTM can be summarized as shown in Figure 6 as follows.

#### *3.3.2. Use ABC's algorithm to optimize LSTM parameters [22].*

In this part, the researcher has divided the work into 6 phases as follows.

Phase 1: Input parameter
Phase 2: Defining Objective and Fitness Function
Phase 3: Generate Initial Population
Phase 4: Perform Employed and Onlooker Phases
  Select variable and Partner
  Generate New Solution
  Perform Greedy Selection
Phase 5: Memorize the Best Selection
Phase 6: Perform Scout Phase

Each phase can be described in detail as follows.

```
#Make MultiLayer LSTM Model
model = keras.Sequential()
model.add(layers.LSTM(32, input_shape=(120, 1), return_sequences=True))
model.add(layers.LSTM(32, return_sequences=True))
model.add(layers.LSTM(32))
model.add(layers.Dense(1))
model.summary()

Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_1 (LSTM)               (None, 120, 32)           4352

 lstm_2 (LSTM)               (None, 120, 32)           8320

 lstm_3 (LSTM)               (None, 32)                8320

 dense_3 (Dense)             (None, 1)                 33

=================================================================
Total params: 21,025
Trainable params: 21,025
Non-trainable params: 0
_________________________________________________________________
```

**Fig. 6 The summary of Long Short-Term Memory for prediction of PM2.5**

*Phase 1: Input Parameter*

The LSTM used in this research uses a multi-layer LSTM, and the optimizer uses ADAM. Therefore, in the process of configuring the parameters, the minimum and maximum values of each layer connection are as follows. The minimum of 4 layer connections, the maximum of 64, and the minimum learning rate of 0.00001, the maximum of 0.01. This means that these parameters are the individual bee members that will be used to determine the optimal value, as detailed in Figure 7.

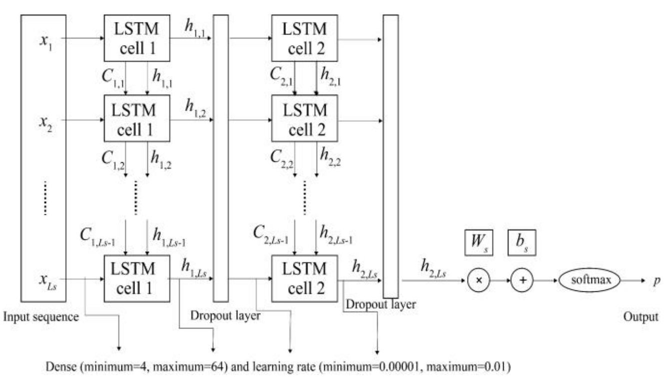



**Fig. 7 The figure shows the initialized parameter and definition of character in bee member for the optimizer of LSTM.**

*Phase 2: Defining Objective and Fitness Function.*

Use the formulas for calculating the accuracy (ACC) and mean square error (MSE) from Equations 6 and 7, respectively, as follows:

$$\boldsymbol{Objective\ function} = \max(\boldsymbol{Acc_{train\ and\ test}})\ \boldsymbol{and}\ \min(\boldsymbol{MSE}) \quad (8)$$

*Phase 3: Generate Initial Population*

Because the population determination of the bee colonies used in this research was based on density randomization and learning rate randomization of the LSTM neural network, it can be expressed as a pseudo-statement as follows.

**Generate of Initial population**
**Start**
*lower bound of dense (lb)= 4, upper bound of dense (ub)=64,*
*lower bound of learning rate (lbr)=0.00001, upper bound of learning rate (ubr)=0.01*
*i=0, j=0, D=3, population size (N)=100*
*while i <= (N/2) do*
*while j <= (D - 1) do*
$\boldsymbol{Y_i^j}$ *= round (lb + random.random() * (ub - lb), 0)*
*j+=1*
*end*
$\boldsymbol{Y_i^3}$*= lbr + random.random() * (ubr - lbr)*
*i+=1*
*end*

**Affinity Evaluation and Sorting**
*For i in range(N/2)*
*Calculate the objective function* ($\boldsymbol{Y_i}$)
*Sort* ($\boldsymbol{Y_i}$) *based on the objective function*
*End For*
**Stop**

*Phase 4: Perform Employed and Onlooker Phases*

In the employee's work process, the bee is to search for a new food source according to Equation 9 and then calculate the suitability according to Equation 8. If the new position value calculated is better, the position will be adjusted according to the new value.

$$\overrightarrow{\boldsymbol{V_{ij}}} = \overrightarrow{\boldsymbol{X_{ij}}} + \emptyset \times \boldsymbol{X_{ij}} - \boldsymbol{X_{rj}} \quad (9)$$

where $\overrightarrow{\boldsymbol{V_{ij}}}$ is the location of the new food source.
$\emptyset$ is random number between [0,1].
$\boldsymbol{X_{ij}}$ is the population.
$\boldsymbol{X_{rj}}$ is the randomized population at r at the jth position.

The work of the onlooker bee determines the food source from the worker bees based on the probability obtained from Equation 10. If the food source is highly probable, the onlooker bee is more likely to be selected. The selected data will be used for calculations to find more suitable food sources similar to Employee bees.

$$\boldsymbol{P_i} = \frac{\boldsymbol{fit_i}}{\sum_{i=1}^{S_n} \boldsymbol{fit_i(X_i)}} \quad (10)$$

Therefore, both phases can be summarized as follows:
Employee Bee Phase
For each food source
Select the update Variable and neighbouring food source
Update the variable with Equation 9
Apply Greedy Selection
probabilities
Onlooker Bee Phase
Select the random number (r)
If r ¡ prob
Update the food sources based on random variables and probability

*Phase 5: Memorize the Best Selection*

In this phase, a certain optimum level is achieved, so the memory stores the value to compare it with the next available value.

*Phase 6: Perform Scout Phase*

The scout bee phase function starts when an employee bee's original food source is not selected from a specified number of onlooker bees; the scout bee will randomly replace the unselected food source with a new food source. In the scout bee phase, the work process can be summarized as follows.

Scout Bee Phase
While food source does not improve
count+=1
If count > $\boldsymbol{count_{max}}$
Replace the food source with the newly generated food source
Else
count=0
Iteration+=1

## 4. Experimental Result

From the workflow of the research methodology part, the researcher started by defining the parameters of the LSTM statically. The data above were used for training and testing at each station by dividing the training and testing. The random method used 70% for training and the remaining 30% for testing. The characteristics of the data are shown in Figure 5. However, for processing the work in this research, the researcher used a workstation computer type Lenovointel core i7 gen $10^{th}$ GPU, speed 2.60 GHz, 32 GB ram, OS Microsoft Windows 11 Pro. In the LSTM processing with fixed parameters, as shown in Figure 6, it was found that the processing time was about 15 minutes, and the total number of working cycles was 150 cycles. The experimental results are shown in Table 1.

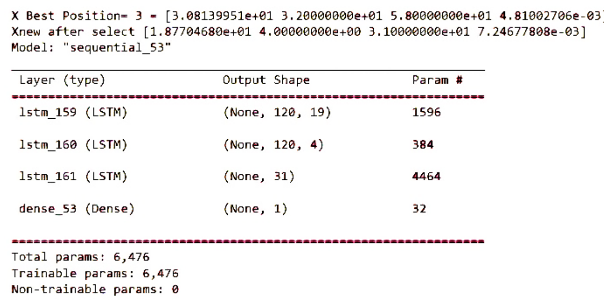
From the experimental results in Table 1, it was found that in determining the constant parameters of the LSTM, as shown in Figure 6, using the ADAM type optimizer, the average training accuracy was 97%. The accuracy by the average of the tests is 96%, which is a fairly high value, and the mean square error (MSE) values are uneven, with the average MSE being 0.00740, which is still a fairly high value. However, the LSTM neural network's accuracy was quite high because the data in each station was not complicated, and the uncertainty was not high. PM2.5 values are quite close to each other; there is not much difference. In addition, the predictive ability of the LSTM neural network performs quite well, but on the other hand, the error value is still high.

Therefore, to reduce the error value, the researcher has an idea to adjust the parameters of the artificial neural network LSTM to be the most suitable for the input PM2.5.

**Table 1. The table summarizes the results of the accuracy and MSE at 7 stations for LSTM (Adam) neural network**

| Station St Number | Train Accuracy | Test Accuracy | Error MSE |
|---|---|---|---|
| ST01 | 97.16060% | 96.85822% | 0.00840 |
| ST02 | 98.16029% | 97.28822% | 0.00728 |
| ST03 | 98.25029% | 96.28530% | 0.00684 |
| ST04 | 97.15811% | 97.28822% | 0.00798 |
| ST05 | 96.16070% | 96.27828% | 0.00809 |
| ST06 | 98.56050% | 97.28705% | 0.00651 |
| ST07 | 97.16850% | 96.89763% | 0.00673 |

From researching relevant research papers, it was found that the Artificial Bee Colony algorithm (ABC algorithm) was suitable to be used to optimize the parameters of the LSTM neural network. The research process is shown in the research method section above. The results obtained by using ABC to adjust the parameters of the LSTM neural network can be summarized in the model, as shown in Figure 8 as follows.


```
X Best Position= 3 = [3.08139951e+01 3.20000000e+01 5.80000000e+01 4.81002706e-03]
Xnew after select [1.87704680e+01 4.00000000e+00 3.10000000e+01 7.24677808e-03]
Model: "sequential_53"

Layer (type)              Output Shape             Param #
==============================================================
lstm_159 (LSTM)           (None, 120, 19)          1596

lstm_160 (LSTM)           (None, 120, 4)           384

lstm_161 (LSTM)           (None, 31)               4464

dense_53 (Dense)          (None, 1)                32

==============================================================
Total params: 6,476
Trainable params: 6,476
Non-trainable params: 0
```

**Fig. 8 The summarized model of optimization using the Artificial Bee Colony algorithm (ABC algorithm) of Long Short-Term Memory for prediction of PM2.5.**

The experimental results using the ABC algorithm for parameter optimization of the LSTM neural network work as follows: the random method used 70% for training and the remaining 30% for testing. The characteristics of the data are shown in Figure 6. Moreover, for processing the work in this research, the researcher used a workstation computer type Lenovointel core i7 gen $10^{th}$ GPU, speed 2.60 GHz, 32 GB ram, OS Microsoft Windows 11 Pro. Training in each work cycle is 10 epochs. The population of 100 bees is divided into 50 best employee bees, and in these 50, 25 onlooker bees are used. The number of generation times is 5 generations—an average total processing time of 1 hour. From the experimental results of the ABC algorithm to find the optimization value, it was found that the optimized parameter that caused the least error can be summarized as a model, as shown in Figure 8. In addition, it can show that the overall number of parameters is less than LSTM neural network with fixed parameters, as shown in Figure 6. Therefore, the experimental result of Figure 8 is that using the overall number of parameters is 6,476 params while using the overall number of parameters before optimization according to Figure 7 is 21,025 params.

**Table 2. The table shows the comparison of the accuracy and error between the LSTM neural network and the parameterized neural network using the ABC algorithm.**

| | Model LSTM (Adam) | | | Optimization Model LSTM (Adam) Using ABC | | |
|---|---|---|---|---|---|---|
| Station | Train | Test | Error | Train | Test | Error |
| St Number | Accuracy | Accuracy | MSE | Accuracy | Accuracy | MSE |
| ST01 | 97.16060% | 96.85822% | 0.00840 | 98.48793% | 96.88799% | 0.00291 |
| ST02 | 98.16029% | 97.28822% | 0.00728 | 98.48758% | 96.88799% | 0.00267 |
| ST03 | 98.25029% | 96.28530% | 0.00684 | 98.48758% | 96.88799% | 0.00272 |
| ST04 | 97.15811% | 97.28822% | 0.00798 | 98.48758% | 96.88799% | 0.00278 |
| ST05 | 96.16070% | 96.27828% | 0.00809 | 98.48758% | 96.88799% | 0.00297 |
| ST06 | 98.56050% | 97.28705% | 0.00651 | 98.48758% | 96.88799% | 0.00281 |
| ST07 | 97.16850% | 96.89763% | 0.00673 | 98.48758% | 96.88799% | 0.00287 |

Moreover, the error effect of using the ABC algorithm for optimizing the parameters of this LSTM neural network was significantly reduced by showing the comparative values before the LSTM neural network. Using the ABC algorithm to optimize the parameters and after using this algorithm to optimize the parameters as shown in Table 2.

The experimental results shown in Table 2 showed that the accuracy obtained from each station adjustment was significantly increased, averaging from training at 98% and from the testing at 96%. In addition, the average mean square error (MSE) was 0.00281. Therefore, it can be concluded that the accuracy obtained from the parameter adjustment with the ABC algorithm was higher, while the error was significantly decreased number accordingly. In addition to the experimental results shown in Table 2, which demonstrate a good elimination result using appropriate algorithm adjustments. To have more confidence, the researcher presents a table showing the comparison of the error of this research with other related or similar studies, as shown in Table 3 below.

**Table 3. The table is a comparison of experimental results from relevant research using the ABC algorithm to optimize parameters for LSTM neural networks.**

| | RMSE | MSE |
|---|---|---|
| Junfu Xi and et. al. [20] | 3.7588 | 14.1287 |
| Raghavendra Kumar and et. al. [21] | | |
| - AAPL | 7.3565 | 54.1180923 |
| - MSFT | 3.8678 | 14.9598768 |
| - INTL | 1.6568 | 2.74498624 |
| Jie Heng and Min Li [22] | 4.63680925 | 21.50 |
| Chen Zhang and et. al. [23] | | |
| - Shanghai | 17.6697 | 312.218298 |
| - Guangzhou | 12.0868 | 146.090734 |
| Wang Qiang and et. al. [24] | 18.5157 | 342.831146 |
| Atiwanwong T and et.al. | 0.05167204 | 0.00267 |

Table 3 shows the relevant research on the use of artificial neural networks and the use of an ABC algorithm to optimize the parameters so that the output values have the minimum errors: JunfuXi et al. [23] ABC algorithm to adjust the parameters of backpropagation neural network (BP) and support vector machine (SVM) to be used for PM2.5, PM10, etc. The researcher results in the output as a measurement of MSE error = 14.1287 and RMSE = 3.7588. Raghavendra Kumar et al. [24] used the LSTM neural network in conjunction with the ABC algorithm to forecast the NASDAQ stock market with 10 years of historical data, including stocks of IT sector funds Apple Inc. (AAPL), and Microsoft Corporation (MSFT). and Intel Corporation (INTL) use the output value to measure the error type RMSE. The result shows that the average error RMSE = 4.2937. Jie Heng et al. presented research that predicts PM2.5 in Mianyang City with a Back Propagation neural network (BP), which is parameterized with the Bee Colony algorithm to obtain the minimum RMSE = 4.6368. Jie Heng et al. [23] presented research that predicts PM2.5 in Mianyang City with a BackPropagation neural network (BP), which is parameterized with the ABC algorithm to obtain the minimum RMSE = 4.6368. Chen Zhang et al. [26] presented research on PM2.5 generation in China in two major cities: Shanghai and Guangzhou. In this work, the researcher uses a multi-fractal dimension image discrimination technique to make the accuracy higher and the error less. The method of the ABC algorithm helps to find the optimal parameters of multifractal dimension and applies the technique of SVM to predict the next day's PM2.5 emissions. Moreover, the results of this research found that the error was 17.6697 for Shanghai City and 12.0868 for Guangzhou City, respectively. Wang Qiang et al. [27] presented research on PM2.5 prediction using a BP neural network combined with the ABC algorithm parameter optimization algorithm. Comparison of the error in prediction before optimization with ABC algorithm and after optimization found that after optimization with ABC algorithm, the error value was minimum, RMSE = 18.5157. Atiwanwong presents the experimental results found that the error values for both RMSE and MSE are less than all the above research (Table 3), i.e. RMSE = 0.05167 and MSE = 0.00267, respectively. The LSTM neural network and its parameters are optimized and give good performance output values resulting from the use of the ABC algorithm accordingly mentioned above.

In this research, the development of the PM2.5 prediction method by using the LSTM neural network was presented to obtain the best output value, which is high accuracy and small error; divided the research method is divided into two parts.

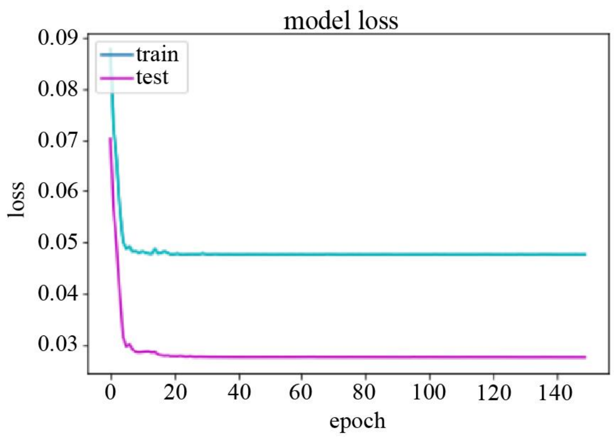



**Fig. 9 Training and Validation Loss of LSTM neural network for constant the parameters at ST01.**

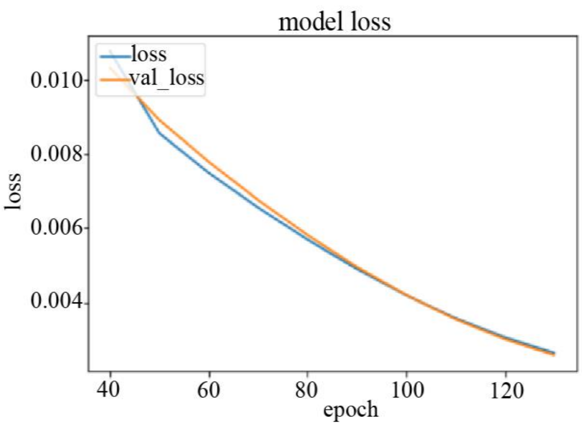



**Fig. 10 Training and Validation Loss of optimization of the parameter of LSTM neural network using ABC algorithm at ST01.**

The first part uses a constant value of parameters related to the neural network. The results from all stations are shown in Table 1. The average accuracy of 96%, an average error (MSE) of 0.00651, and the average number of cycles to work at 150 epochs and measuring the model loss value will be shown in Table 2. As can be seen from Figure 9, overfitting is an indication that the model is not fit. An overfit occurs when the model is unable to model the training data properly. This suggests that additional training is required to reduce the losses incurred during training or that misconfiguration of the parameters has led to this event which, if applicable, may make predictions more inaccurate. To address this overfitting problem, in the second part, this paper uses a heuristic algorithm to optimize the LSTM neural network's parameters for the best model loss result. As for the heuristic algorithm used, this research chose to use the ABC algorithm because of the merits and necessity the researcher mentioned in the introduction and related work.

Experimental results obtained using the ABC algorithm to optimize LSTM neural network parameters showed that model loss had a good fitting effect. The training loss and validation loss decreased and stabilized at a specific point, as shown in Figure 10. Figure 10 shows that when using the ABC algorithm to adjust the parameters, the good fit result is higher accuracy, less error, the number of epochs (10 epochs), and the sum of all parameters. Significantly less, with a mean accuracy of 98% and a mean error (MSE) of 0.00267, and compared to other studies in Table 3, it was found that the effect of using the adjustment of the parameters in this research clearly outperformed. When the prediction results obtained from the experiment were compared to the values obtained from the actual data collection, it was found that the graph results were almost the same line, showing the performance in predicting PM2.5 at the maximum accuracy while the error value is less, as shown in Figure 11.

When the results of each predicted station were compared, it became clear that the error at each station obtained from the parameterization of the LSTM neural network was lower than that of the expert in the LSTM neural network. The configuration may cause overfitting or underfitting, which will further affect the prediction performance. Figure 12 shows a graph comparing the performance of the LSTM neural network with parameterization vs using the experts' fixed parameterisation.

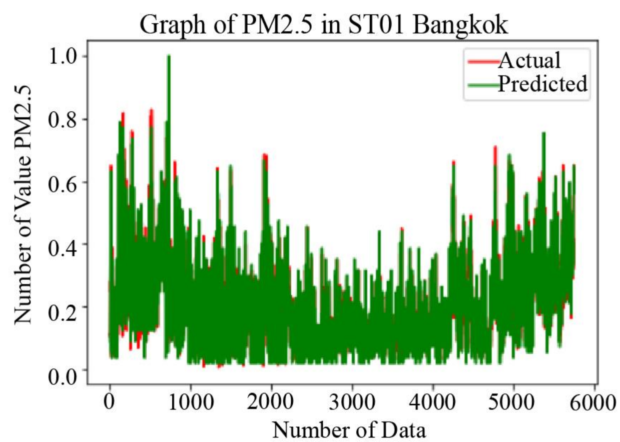



**Fig. 11 The graph shows the results of the experimental predictions compared to the actual data at ST01.**

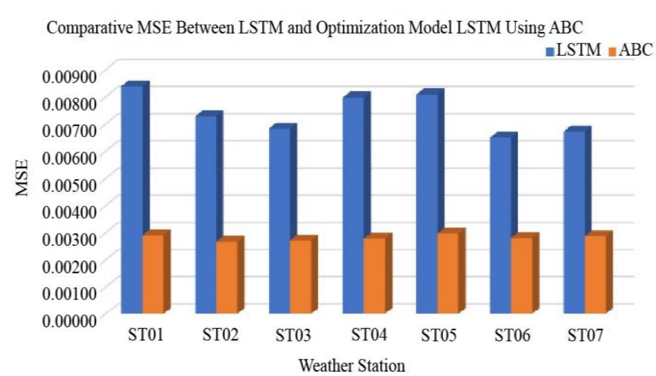



**Fig. 12 The graph shows the results of the MSE values compared between the LSTM neural network and the optimized parameter of the LSTM neural network.**

From the experimental results, it was found that the LSTM neural network was able to predict PM2.5 effectively, especially on complex and time-series data. Therefore, it can be concluded that the LSTM neural network has good performance in predicting PM2.5 values. However, to make it more efficient, it is necessary to adjust the parameters to suit the input data. Trains and tests have an effect on the prediction as well as overfitting and underfitting. For good fitting, it is important to consider tuning all parameters, especially those of the hyperparameter, to obtain the best possible prediction.

## 5. Conclusion

This research presents the prediction of PM2.5 occurrence using the Long Short Term Memory neural network and adjusts parameter optimization with the Artificial Bee Colony algorithm. The accuracy result when using LSTM was 96%, while the mean square error was 0.00651, the total parameter of 21,025 params considered a good value. But when the LSTM neural network was used to adjust the parameters to be suitable for the input values by using hyperparameter tuning with the Artificial Bee Colony algorithm, it was found that both the accuracy and the error values were better output values. The average accuracy was 98%, the total parameter was 6,476 params, and the average error was 0.00267. It shows that the LSTM neural network is most suitable for predicting PM2.5. If the appropriate algorithm is used to adjust the hyperparameter to the input value, it will result in the best export value for the PM2.5 prediction obtained, respectively.

## 6. Recommendation

The aforementioned research has shown that a parameter-optimized LSTM neural network can predict the occurrence of PM2.5 more effectively than a specialist parameterization. In particular, selecting the appropriate algorithm is also important in tuning the parameters of the LSTM neural network. However, predicting the occurrence of PM2.5 dust alone may not be sufficient for protection. Therefore, forecasting is necessary to proceed from prediction. In order to make research related to the prevention of PM2.5 occurrence more complete in future research, the researcher has a guideline for conducting research which is to use regression equations for forecasting only the optimization of the parameters of the equation has the ability to predict the highest accuracy. Ultimately, this upcoming research will combine the optimization of the LSTM neural network's parameters and the regression equation's optimization variable. To make the prediction, the output value with the highest accuracy and the smallest error value will appear in future research.

## References

[1] T. Airveda, What Is Pm2.5 and Why Is It Important?, Airveda, 2017. [Online]. Available: https://www.airveda.com/blog/what-is-pm2-5-and-why-is-it-important

[2] *Ambient Air Pollution,* World Health Organization: Health Impact, Institute for Health and Evaluation, University of Washington, 2020.

[3] Yanlin Qi et al., "A Hybrid Model for Spatiotemporal Forecasting of PM2.5 Based on Graph Convolutional Neural Network and Long Short-Term Memory," *Science of the Total Environment*, vol. 664, pp. 1-10, 2019. *Crossref*, https://doi.org/10.1016/j.scitotenv.2019.01.333

[4] Dewan Seng et al., "Spatiotemporal Prediction of Air Quality Based on LSTM Neural Network," *Alexandria Engineering Journal,* vol. 60, no. 2, 2020. *Crossref*, http://dx.doi.org/10.1016/j.aej.2020.12.009

[5] Thanpitcha Atiwanwong, and Saweth Hongprasit, "A Low-Power Real-Time Pollution Monitoring System Using ESP LoRa," *Mahasarakham International Journal of Engineering Technology*, vol. 6, no. 1, pp. 36–40, 2020. *Crossref*, https://doi.org/10.14456/mijet.2020.8

[6] Syed Ahsin Ali Shah et al., "A Hybrid Model for Forecasting of Particulate Matter Concentrations Based on Multiscale Characterizations and Machine Learning Techniques," *Mathematical Biosciences and Engineering,* vol. 18, no. 3, pp. 1192–2009. *Crossref*, https://doi.org/10.3934/mbe.2021104

[7] Rayner, and David Christopher Ferguson, "*Optimization for Heuristic Search*," Thesis ERA, 2014, *Crossref*, https://doi.org/10.7939/R39W91

[8] Vincent Kenny, Matthew Nathal, and Spencer Saldana, Heuristic Algorithms, Optimization, 2014, [Online]. Available: https://optimization.mccormick.northwestern.edu/index.php/heuristic/algorithms

[9] Feng Jiang et al., "Atmospheric PM2.5 Prediction Using Deepar Optimized by Sparrow Search Algorithm with Opposition-Based and Fitness-Based Learning," *Atmosphere,* vol. 12, no. 7, pp. 1–18, 2021. *Crossref,* https://doi.org/10.3390/atmos12070894

[10] Jie Heng, and Min Li, "Prediction of PM 2.5 Concentration Based on BP Neural Network Optimized By Bee Colony Algorithm," *MATEC Web of Conferences,* vol. 355, p. 03025, 2022. *Crossref,* https://doi.org/10.1051/matecconf/202235503025

[11] Sang Won Choi, and Brian H. S. Kim, "Applying PCA to Deep Learning Forecasting Model for Predicting PM2.5," *Sustainability,* vol. 13, no. 7, pp. 1–30, 2021. *Crossref,* https://doi.org/10.3390/su13073726

[12] Qiang Wang et al., "PM2.5 Prediction Model Based on ABC-BP," *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pp. 140–143, 2019. *Crossref,* https://doi.org/10.1109/CISCE.2019.00039

[13] X.-S. Yang, Nature-Inspired Optimization Algorithms, 2014, [Online]. Available: https://www.sciencedirect.com/science/article/pii/b9780124167438000166

[14] B. S. Chaudhari, and M. Zennaro, LPWAN Technologies for IoT and M2M Applications, Academic Press, pp. 1-3, 2020, [Online]. Available: https://www.sciencedirect.com/science/article/pii/b978012818880400020x

[15] *Health and Envitech*, Air Quality Problems from PM2.5 in Thailand, Air Quality Problems from PM2.5, 2022, [Online]. Available: https://www.https://healthenvi.com/

[16] Sepp Hochreiter, and Jürgen Schmidhuber, “Long Short Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. *Crossref*, https://doi.org/10.1162/neco.1997.9.8.1735

[17] M. Dhanalakshmi, and V. Radha, "Discretized Linear Regression and Multiclass Support Vector Based Air Pollution Forecasting Technique," *International Journal of Engineering Trends and Technology*, vol. 70, no. 11, pp. 315-323, 2022. *Crossref*, https://doi.org/10.14445/22315381/IJETT-V70I11P234

[18] S. Neelima et al., “A Comprehensive Survey on Variants in Artificial Bee Colony (ABC),” *International Journal of Computer Science and Information Technologies*, vol. 7, no. 4, pp. 1684-1689, 2016.

[19] Palak, Preeti Gulia, and Nasib Singh Gill, “Optimized Test Case Selection Using Scout-Less Hybrid Artificial Bee Colony Approach and Crossover Operator,” *International Journal of Engineering Trends and Technology*, vol. 69, no. 3, pp. 39-45, 2021. *Crossref*, 10.14445/22315381/IJETT-V69I3P208

[20] Karaboga, Dervis, and Celal Ozturk, “A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm,” *Applied Soft Computing* vol. 11, pp. 652–657, 2011.

[21] U. Taetragool, “Adaptive Hyperparameter Using Artificial Bee Colony Algorithms,” Big Data Experience Center, 2021. [Online]. Available: http://bigdataexperience.org/hyper-parameter-with-bee-colony/

[22] R. K. Punitha Stephan, Thompson Stephan, and A. Abraham, “A Hybrid Artificial Bee Co Algorithm for Improved Breast Cancer Diagnosis,” *Neural Computing and Applications*, vol. 33, no. 2, pp. 13667–13691, 2021. *Crossref*, https://doi.org/10.1007/s00521-021-05997-6

[23] Junfu Xi et al., “Bp-Svm Air Quality Combina Gression,” *International Journal of Earth Sciences and Engineering*, vol. 9, no. 3, pp. 1194–1199, 2016.

[24] P. K. Raghavendra Kumar, and Y. Kumar, “Integrating Big Data Driven Sentiments Polarity and ABC-Optimized LSTM for Time Series Forecasting,” *Multimedia Tools and Applications*, vol. 81, pp. 34595–34614, 2021. *Crossref*, https://doi.org/10.1007/s11042-021-11029-1

[25] J. Heng, and M. Li, “PM2.5 Prediction Based on Bp Neural Network Optimized By Bee Colony Algorithm,” *ICPM2021*, vol. 30250, pp. 1–8, 2022.

[26] Chen Zhang and et al., “PM2.5 Prediction Base on Multifractal Dimension and Artificial Bee Colony Algorithm,” *Journal of Physics Conference Series*, vol. 1237, no. 2, pp. 1–8, 2019. *Crossref*, https://doi.org/10.1088/1742-6596/1237/2/022085

[27] Qiang Wang et al., “PM2.5 Prediction Model Based on ABC-BP,” *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pp. 141–143, 2019. *Crossref*, https://doi.org/10.1109/CISCE.2019.00039

[28] Md. Mostafizer Rahman et al., “A Bidirectional LSTM Language Model for Code Evaluation and Repair,” *Symmetry*, vol. 13, no. 247, pp.1–15, 2021. *Crossref*, https://doi.org/10.3390/sym13020247