Dispersion Model of Volatile Organic Compounds Based on RBF Neural Network

Liu Anqi¹,², Dongfeng Zhao¹

¹Center for Safety, Environmental & Energy Conservation Technology of China University of Petroleum
²CNPC Research Institute of Safety and Environment Technology
Liu Anqi: liuanqi@cnpc.com.cn

Abstract. In this paper, we try to establish the dispersion model of Volatile Organic Compounds (VOCs). Whereas the mechanism dispersion model is too complicated and only suitable for the case where only one release source exists, we establish the black-box model based on data for the case where multiple release sources exist. After the input and output variables are determined, the Radial-Basis Function (RBF) neural network is used to describe the relationship between the inputs and the outputs because it can approximate arbitrary function with arbitrary precision. Considering the difficulty in selecting the parameters of RBF neural network, we use the Swarm Cuckoo Search (SCS) algorithm to obtain the suitable parameters. The simulation experiments demonstrate the effectiveness of the proposed model.

1. Introduction
Volatile Organic Compounds (VOCs) commonly exist in chemical industries and can cause serious environmental problems, such as greenhouse effect, ozone layer depletion and photochemical smog [1-3]. In petroleum refinery, the VOCs released from the refining devices is one of the major air pollution sources [4]. Therefore, it is time to take some actions to reduce the amount of VOCs in the atmosphere. The current actions taken for controlling the VOCs in petroleum refinery fall into three types, namely source control, supervisory control and tail gas control. The source control addresses the dynamic/static seal points where VOCs release. The dynamic/static seal points may exist in many devices like pumps, valves, pipelines etc. Therefore, the source control is to reduce the leak at the root by using advanced technical process or selecting equipment and material strictly. Supervisory control is to programmatically manage the technical processes which release VOCs probably. Tail gas control is to prevent the VOCs in the tail gas from coming into the atmosphere by burning or absorption. In this paper, we mainly consider the supervisory control.

According to the estimation of American environmental protection agency, the VOCs leaking from the refining devices accounts for a high share of the total emissions, about 0.01%-0.02%. These unorganized releases may be due to the abnormalities of the refining devices. Therefore, the supervising of the refining devices is necessary. However, the quantity of the potential release source is undoubtedly huge. The cost is huge if we install a detection device for each potential release source. Therefore, we can first install some detection devices and then infer the release source based on detection results. To this end, we need to establish a dispersion model.

Currently, some air quality models have been proposed and used to infer the release sources [5-7]. Among them, Gaussian plume model is a popular model [8-10]. Gaussian plume model is a mechanism
model, which results in its wide application. However, the limitation of Gaussian plume model is that it only considers the case where only one release source exists. In practice, the case where multiple release sources exist is commonly encountered.

The mechanism model for the case where multiple release sources exist has not been well studied and with no doubt is a complex problem. Hence, the black-box model established based on data plays an important role. Neural network is an effective tool for establishing black-box model because it can approximate arbitrary function with arbitrary precision [11, 12]. Neural network has many types, like BP neural network, RBF neural network etc. RBF neural network has the advantages of fast training and avoiding local optimum [13]. However, it still has some problems, such as how to determine its center vectors and widths. Swarm intelligent optimization algorithm (SIOA) can deal with complex optimization problem as it does not need strict goal function. Swarm cuckoo search algorithm, as one kind of SIOA, has a strong ability of global convergence [14]. Therefore, in this paper, we try to establish the black-box dispersion model of VOCs by using RBF neural network. Swarm cuckoo search algorithm is used to train the RBF neural network.

2. Factors Affecting Dispersion

The black-box model is established based on the data of the sample inputs and sample outputs. Although its universality is inferior to the mechanism model, it is much easier to obtain. Whereas the inputs and the outputs determine the corresponding black-box model, we should determine the input and the output variables before modeling.

The output variable is easy to determine. We usually use the concentration of VOCs in one spot as the output variable. \( C \) (mg/m\(^3\)) denotes the concentration. According to this output, we will determine the input variables, which are the factors affecting dispersion. The sketch map of dispersion is shown in figure 1.

The first factor is the characteristics of the release source, including the intensity and height of the release source. \( Q \) (mg/m) denotes the intensity and \( H \) (m).

Apparentely, wind is another important factor. Whereas the dispersion occurs downwind, we only consider the effect of the wind speed. \( V \) (m/s) denotes the wind speed.

The different spots downwind may have different concentrations. Therefore, the relative location between the detecting spot and the release source is also a factor. Generally, we can set up a space rectangular coordinates. The coordinate of the release source is \((0, 0, H)\) and the coordinate of the detecting spot is denoted as \((x, y, z)\).

Atmospheric stability can also affect the dispersion. According to different weather conditions, the atmospheric stability has six cases listed in Table 1[15]. In Table 1, A, B, C, D, E, F denote extremely unstable, moderately unstable, slightly stable, neutrally stable, slightly stable and moderately stable respectively. Here, we use \( S \) to denote the atmospheric stability. A–F are assigned 1–6 respectively.

| Wind speed (m/s) | Daytime insolation | Nighttime conditions |
|-----------------|-------------------|---------------------|
|                 | Strong            | Moderate            | Slight            | Low cloud | Cloudiness |
| <2              | A                 | A-B                 | B                 | F         | F          |
| 2-3             | A-B               | B                   | C                 | E         | F          |
| 3-4             | B                 | B-C                 | C                 | D         | E          |
| 4-6             | C                 | C-D                 | D                 | D         | D          |
| >6              | C                 | D                   | D                 | D         | D          |
Generally, ground conditions are also considered as a factor when mechanism model is established. However, the black-box model corresponds to a concrete object. Hence, the ground conditions will be regarded as invariables. The ground conditions will not be inputs. When the inputs and outputs are determined, we can establish the corresponding black-box model.

3. Dispersion Model Based on RBF Neural Network

3.1. RBF Neural Network

The basic structure of RBF neural network is shown in Figure 2. It is a feedforward neural network with a single hidden layer. Although its structure is simple, it has the advantages of fast training, global optimum.

![Figure 2. Basic structure of RBF neural network](image)

In Figure 2, $u$ is the input vector, $u = [u_1, u_2, \ldots, u_m]$; $y$ is the output vector, $y = [y_1, y_2, \ldots, y_n]$; $q$ is the output vector of the hidden layer, $q = [q_1, q_2, \ldots, q_r]$. The transformation from input vector to output vector is realized based on the hidden layer. The relationship between $q$ and $u$ is

$$q_k = R_k([u - c_k])$$  \tag{1}

where $c_k$ is the center vector of the $k$th node in hidden layer; $\| \cdot \|$ is chosen as Euclidean norm; $R_k(\cdot)$ is the radial-basis function, which is key of implementing nonlinear mapping. One-dimensional gaussian radial-basis function (shown as Equation 2 and Figure 3) is usually used, where $\sigma_k$ is the width of $R_k(\cdot)$.

$$R_k(x) = \exp \left( -\frac{x^2}{2\sigma_k^2} \right)$$  \tag{2}

![Figure 3. One-dimensional gaussian RBF](image)

The relationship between $y$ and $q$ is

$$y_i = \sum_{k=1}^{r} \omega_k q_k - \theta_i$$  \tag{3}

where $\omega_k$ is the weight between $y_i$ and $q_k$; $\theta_i$ is the threshold value of $y_i$. Then, based on sample input $u^p$ and sample output $d^p$, $p = 1, 2, \ldots, l$, we can train the neural network to get the optimal parameters. The objective function is

$$J = \frac{1}{2} \sum_p \| d^p - y^p \|^2$$  \tag{4}
where \( y^p \) is the output of the neural network when the input is \( u^p \). Our target is to realize Equation (5) by optimizing parameters, where \( \varepsilon \) is a specified upper limit.

\[
J \leq \varepsilon \tag{5}
\]

According to Equation (3), obviously, the relationship between \( y \) and \( q \) is linear. The tuning of weights and threshold values is equivalent to a linear optimization problem. This problem can be solved based on recursive least square (RLS) [16]. However, the relationship between \( q \) and \( u \) is nonlinear. It is still a difficult problem to determine the center and width of the radial-basis function. In this paper, we will use swarm cuckoo search (SCS) algorithm to determine the parameters of RBF neural network.

### 3.2. Swarm Cuckoo Search (SCS) Algorithm

Swarm cuckoo search algorithm is the modification of the basic cuckoo search algorithm. The basic cuckoo search algorithm is proposed based on the breeding behavior of cuckoo [17]. The procedure of basic cuckoo search algorithm is listed as follows:

**Step 1:** Initialize the search parameters and solution set. The search parameters which need initialization are population \((z)\), step length coefficient \((a)\), discovery probability \((p_a)\), recursion terminal condition (maximum generation or tolerable error). Meanwhile, an initial solution set, \(i = 1, 2, \ldots, z\) should be provided.

**Step 2:** Generate a new solution set \(X_i^{new}, i = 1, 2, \ldots, z\). The new solution set will be generated based on Levy flight:

\[
x(t+1) = x(t) + a \times \text{Levy}(\lambda) \tag{6}
\]

where \( x(t) \) is the current solution; \( x(t+1) \) is the next generation solution; \( a \) is step length coefficient; \( \text{Levy}(\lambda) \) denotes a random step length which obey Levy distribution (shown as Equation 7)

\[
\text{Levy}(\lambda) \sim \mu t^{-\lambda}, 1 < \lambda \leq 3 \tag{7}
\]

**Step 3:** Determine fitness function and calculate fitness values. A proper fitness function needs to be determined. Then, calculate the fitness values of the initial solution set \(F_i^{old}, i = 1, 2, \ldots, z\). Note down the best fitness value and corresponding solution.

**Step 4:** Calculate the fitness values of the new solution set \(F_i^{new}, i = 1, 2, \ldots, z\). If \(F_i^{new}\) is superior to \(F_i^{old}\), a random probability \(p\) will be generated. And then if \(p > p_a\), \(F_i^{old}\) will be replaced by \(F_i^{new}\). If not, \(F_i^{old}\) will hold.

**Step 5:** Search the optimal solution in the current generation. If the recursion terminal condition is not satisfied, go to Step 2.

The above is the basic cuckoo search algorithm. If the proper search parameters are given, the basic cuckoo search algorithm is capable of finding the optimal solution. However, the basic cuckoo search algorithm has a poor efficiency when the search scale is spacious. In view of this, swarm cuckoo search algorithm is proposed.

Compared with the basic cuckoo search algorithm, swarm cuckoo search algorithm has the following modifications:

1. Whereas the nonconvex optimization problem may have multiple optimal solutions, in swarm cuckoo search algorithm, the optimal solutions in every generation are all noted down and the final solution is the optimal one among the noted solutions.

2. In order to improve the search efficiency, the update strategy of the solution set (Equation 6) is modified as

\[
x_i(t+1) = x_i(t) + \xi \times a(t) \times \text{Levy}(\lambda) + \zeta \times a(t) \times (x_{\text{best}} - x_i(t)) + \varsigma \times a(t) \times (x_{\text{best}}^h - x_i(t)) \tag{8}
\]

where \( \xi, \zeta, \varsigma \) and \( \varsigma \) are coefficients; \( x_{\text{best}} \) and \( x_{\text{best}}^h \) are the global optimal values and local optimal solutions of the \( h \)th computation, respectively. Obviously, the currently optimal solutions will have effects on the new solutions.
3. When the search scale is spacious, the fixed step length coefficient \( a \) will reduce the search efficiency. Therefore, an adaptive step length coefficient is used:

\[
a(t) = a_{\text{max}} - \frac{t}{\text{Max\_Gen}}(a_{\text{max}} - a_{\text{min}})
\]

(9)

where \( a_{\text{max}} \) and \( a_{\text{min}} \) are the predetermined maximum and minimum of \( a \); \( \text{Max\_Gen} \) denotes the maximum generation.

4. In the basic cuckoo search algorithm, the discovery probability \( p_a \) is invariable. If the initial \( p_a \) is inappropriate, the effect of the algorithm will be deteriorated. Therefore, in swarm cuckoo search algorithm, the discovery probability is adaptive and related to the fitness values of the solutions. The adaptive discovery probability is calculated as follows.

Define \( P_x(i) \) as the adaptability of the \( i \)-th solution. \( P_x(i) \) is expressed as

\[
P_x(i) = \sum_{i=1}^{n} \left( \frac{f(i) - f_{\text{min}}}{\sum_{i=1}^{n} (f(i) - f_{\text{min}})} \right), i = 1, 2, ..., n
\]

(10)

where \( f(i) \) is the fitness value of the \( i \)-th solution; \( f_{\text{min}} \) is the minimal fitness value in each generation. Then, the discovery probability can be obtained as

\[
p_a(i) = \frac{p_x(i)}{2 \max(p_x)}, i = 1, 2, ..., n
\]

(11)

According to the breeding behavior of cuckoo, other birds have a probability of discovering the cuckoo’s eggs when the cuckoo lays eggs in other birds’ nests. This is the discovery probability. In the basic cuckoo search algorithm, the old solution has a probability (discovery probability) of not being replaced even if it is inferior to the new solution. Similarly, even if the cuckoo cannot take a nest, its eggs still have a probability of living. Therefore, swarm cuckoo search algorithm introduces the survival probability, which means even if the new solution is inferior to the old one, it still has a probability of replacing the old one. The survival probability \( p_s \) is expressed as

\[
p_s = k_s p_a
\]

(12)

where \( k_s \) denotes the survival coefficient. The value range of \( k_s \) is [0, 1].

Based on the above modifications, the flow chat of swarm cuckoo search algorithm is shown as Figure 4.

We will employ swarm cuckoo search algorithm to determine the parameters of the RBF neural network. The parameters include the centers of radial-basis functions, \( c = [c_1, c_2, ..., c_r] \); the widths of radial-basis functions, \( \Delta = [\sigma_1, \sigma_2, ..., \sigma_r] \); the weight \( \Omega = [\omega_1, \omega_2, ..., \omega_k] \); and the threshold value \( \theta \). The solution which we search for can be determined as \( x = [c, \Delta, \Omega, \theta] \). The fitness function can be determined as Equation (4). Then, we can train the RBF neural network according to Figure 4.

3.3. Dispersion Model

Now, we will establish the RBF neural network model. Firstly, we only consider one release source. As we have analyzed the inputs and outputs of the neural network in section II, the neural network is easy to establish. Figure 4 is the RBF neural network model when a single release source exists.
In Figure 4, the inputs and the output have been determined. The number of the nodes in the hidden layer should also be given. Generally, increasing the number of node can reduce approximate error, but also can complicate the structure of network. Therefore, the initial number of node is 10. The MSE value expressed as Equation (13) is taken as an index.

\[
MSE = \frac{1}{n} \sum_{p=1}^{n} \left[ d^p - y^p \right]^2
\]

where \(y^p\) is the output of the neural network model; \(d^p\) is the sample output. If the MSE is too large, then, increase the number of node. The number of node can keep increasing until the MSE is small enough or tends to a constant.

When multiple release sources exist downwind, the dispersion model will be more complicated. Intuitively, the information of all the release sources should be included in the inputs of the RBF neural network. If the number of release source is large, the RBF neural network will have too many inputs accordingly. The RBF neural network will be too complicated, which is difficult to train. Hence, we will find another way to establish the RBF neural network model when multiple release sources exist.

Assume that there are \(n\) release sources downwind. Obviously, the concentration of the detecting spot \(C\) will be affected by all the release sources. If \(C_i\) denotes the concentration of the detecting spot when only the \(i\)th release source exist, the real concentration \(C\) can be expressed as

\[
C = \Gamma(C_1, C_2, \cdots, C_n)
\]

where \(\Gamma(\cdot)\) denotes the functional relationship between \(C\) and \(C_i\). As this relationship may be too complicated, we can use another RBF neural network to approximate it. Then, the structure of RBF neural network is shown as Figure 6. In Figure 6, the \(n\) RBF neural networks in the first layer are used to obtain...
the concentrations $C_1, C_2, \ldots, C_n$. The RBF neural network model for a single release source can be used directly. Therefore, the computation will be reduced greatly. The last RBF neural network is used to approximate the relationship, $\Gamma(\cdot)$.

It is important to note that the RBF neural network for multiple release sources has a low universality. Generally, if the number of release source changes, the RBF neural network model needs to reestablish.

Figure 6. RBF neural network model when multiple release sources exist

4. Model Validation

In this section, we will verify the effectiveness of the proposed RBF neural network model. Firstly, we will establish the RBF neural network model for a single release source. We have 300 sets of sample data. We select 150 sets for training and the other 150 sets for verification. The initial number of node in hidden layer is 10. Then we train the RBF neural network based on swarm cuckoo search algorithm. The relationship between the number of node in hidden layer and MSE of training data is shown as Figure 7. Determine the number of node as 85 and then MSE satisfy MSE< 1%.

Figure 7. Relationship between MSE and the number of node

Then, we will use the other 150 sets to verify the effectiveness of the RBF neural network. According to the sample inputs, the RBF neural network model can predict the output (concentration). The outputs of the RBF neural network model and the real values are plotted in Figure 8. In Figure 8, the outputs of the model are close to the real values. The MSE of verification data is 2.25%. Therefore, the RBF neural network model for a single release source is effective.

Then, we will establish the RBF neural network model for multiple release sources. Whereas the RBF neural network model for a single release source has been obtained, the first $n$ RBF neural networks can
be replaced by the well-trained RBF neural network directly. Therefore, what we only need to do is to train the last one RBF neural network.

We also obtain 300 sets of sample data. Different from the previous data, these data come from three different release sources which release VOCs at the same time. We also select 150 sets for training and the other 150 sets for verification. Figure 9 shows the comparison between real values and outputs of model. Similarly, the outputs of the model are close to the real values. The MSE of verification data is 4.74%. Therefore, the RBF neural network model for multiple release sources is effective.

In fact, in this example, apart from the cascaded structure of the RBF neural network, namely Figure 6, the structure in Figure 4 can also be used for modeling. If we use the structure in Figure 4, it will cost much more time to train the neural network because the neural network has too many parameters to determine. If the number of release source is too large, it is difficult to use the structure of Figure 4 to establish a RBF neural network model.

Figure 8. Comparison between real values and outputs of model for a single release source

Figure 9. Comparison between real values and outputs of model for multiple release sources

5. Conclusion
This paper presents a method for establishing the dispersion model of Volatile Organic Compounds (VOCs). Whereas the mechanism dispersion model is not suitable for the case where multiple release sources exist, we try to establish a black-box model by using RBF neural networks. Considering the difficulty in determining the parameters of RBF neural networks, swarm cuckoo search algorithm is employed to search for optimal parameters. Meanwhile, a cascaded structure of the RBF neural network is proposed. This structure can reduce the number of undetermined parameters of the RBF neural networks and is suitable for the case where many release sources exist. The examples demonstrate the effectiveness of the proposed models.
References
[1] Geng F, Tang Z X, Lu G, et al. Analysis of ozone and VOCs measured in Shanghai: A case study[J]. Atmospheric Environment, 2007, 41(5):989-1001.
[2] Lee S C, Chiu M Y, Ho K F, et al. Volatile organic compounds (VOCs) in urban atmosphere of Hong Kong.[J]. Chemosphere, 2002, 48(3):375-382.
[3] Hatfield M L, Hartz K E H. Secondary organic aerosol from biogenic volatile organic compound mixtures[J]. Atmospheric Environment, 2011, 45(13):2211-2219.
[4] Huang C , Chen C H , Li L , et al. The study of emission inventory on anthropogenic air pollutants and VOC species in the Yangtze River Delta region, China[J]. Atmospheric Chemistry & Physics Discussions, 2011, 11(1):951-983..
[5] Banerjee T , Barman S C , Srivastava R K . Application of air pollution dispersion modeling for source-contribution assessment and model performance evaluation at integrated industrial estate-Pantnagar[J]. Environmental Pollution, 2011, 159(4):865-875.
[6] J. Carlos García-Díaz, Gozalvez-Zafrilla J M . Uncertainty and sensitive analysis of environmental model for risk assessments: An industrial case study[J]. Reliability Engineering & System Safety, 2012, 107(4):16-22.
[7] Venegas L E . An urban air pollution model[J]. Energy & Buildings, 1991, 16(1-2):705-709.
[8] Islam M A. Application of a Gaussian Plume Model to Determine the Location of an Unknown Emission Source[J]. Water Air & Soil Pollution, 1999, 112(3-4):241-245..
[9] Sykes R I, Lewellen W S, Parker S F. A Gaussian Plume Model of Atmospheric Dispersion Based on Second-Order Closure.[J]. Journal of Applied Meteorology, 1986, 25(25:3):322-331.
[10] Green A E S, Singhal R P, Venkateswar R. Analytic Extensions of the Gaussian Plume Model[J]. Air Repair, 1980, 30(7):773-776.
[11] Cybenko G. Approximation by superpositions of a sigmoidal function[J]. Mathematics of Control Signals & Systems, 1989, 2(4):303-314.
[12] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators.[J]. Neural Networks, 1989, 2(5):359-366.
[13] Wang X, Shao H. The theory of RBF neural network and its application in control [J]. Information and Control, 1997, 26(4):272-284..
[14] Jin Q , Wang Q , Qi L , et al. Swarm cuckoo search for closed-loop parameter identifications from different input signals[J]. The Canadian Journal of Chemical Engineering, 2016,94(9):1755-1769.
[15] Gifford F A. Use of Routine Meteorological Observations for Estimating Atmospheric Dispersion[J]. Nuclear Safety, 1961, 2:47-57.
[16] Feng D, Liu X, Liu M. The recursive least squares identification algorithm for a class of Wiener nonlinear systems[J]. Journal of the Franklin Institute, 2016, 353(7):1518-1526..
[17] Yang X S , Deb S . Cuckoo Search via Levy Flights[J]. Mathematics, 2010:210 - 21