Soft measurement of dioxin emission concentration based on deep forest regression algorithm

Tang Jian, Xia Heng*, Qiao Junfei and Guo Zihao

Faculty of Information Technology,
Beijing University of Technology,
Beijing, 100024, China

and

Beijing Key Laboratory of Computational Intelligence and Intelligent System,
Beijing, 100124, China

and

Beijing Laboratory of Smart Environmental Protection,
Beijing, 100124, China
Email: Xia_heng1220@163.com
Email: freeflytang@126.com
Email: junfeiq@bjut.edu.cn
Email: 1205947202@qq.com
*Corresponding author

Abstract: Dioxin (DXN) is an organic pollutant emitted by the municipal solid waste incineration (MSWI) process. In industrial process, the DXN emission concentration is detected by using offline laboratory analysis method with a monthly/seasonal or un-determined period. In this paper, a soft measurement method of DXN based on deep forest regression (DFR) algorithm is proposed. First, the input layer forest model consists of multiple sub-forest models is trained and the layer regression vector is obtained. Then, the augmented layer regression vector that serial combine the layer regression vector with the raw features is used to train the middle layer forest model. Finally, the augmented layer regression vector of the middle layer forest is fed into the output layer forest model to produce the final DXN prediction. The effectiveness of the proposed method was verified by the benchmark data and the DXN emission concentration data of the actual MSWI process.

Keywords: dioxin; DFR; deep forest regression; layer regression vector; augmented layer regression vector; MSWI; municipal solid waste incineration.

Reference to this paper should be made as follows: Jian, T., Heng, X., Junfei, Q. and Zihao, G. (2021) 'Soft measurement of dioxin emission concentration based on deep forest regression algorithm', *Int. J. System Control and Information Processing*, Vol. 3, No. 3, pp.208–228.

Biographical notes: Tang Jian research interests include small sample data modeling and intelligent control of municipal solid waste treatment process.

Xia Heng research interests include ensemble learning, small sample modeling, and intelligent control of urban solid waste treatment processes.

Qiao Junfei research interests include intelligent control of sewage treatment process, neural network structure design and optimisation.

Guo Zihao research interests include feature modeling of high-dimensional small sample data, and soft measurement of difficult-to-measure parameters in solid waste treatment processes.

This paper is a revised and expanded version of a paper entitled 'Soft measurement of dioxin emission concentration based on deep forest regression algorithm' presented at *The 31st Chinese Process Control Conference*, Xuzhou, China, 30 July – 1 August, 2020.

1 Introduction

Municipal solid waste (MSW) continues to increase at rate of 8–10%/year in the world. This result has caused many cities to face the garbage siege crisis (Zhang et al., 2011). MSW incineration (MSWI) is one of the typical treatment methods to achieve waste reduction, recycling, and harmlessness jointly (Li et al., 2016). At present, the number of MSWI-based power plants in China mainland is increasing year by year, of which the mechanical grate-based MSWIs account for the largest proportion (Qiao et al., 2020). At the same time, as the rapid development of MSWI technology, there are still problems such as lack of operating experience and pollution supervision measures. Thus, it is difficult to meet the pollution emission standards (Lu et al., 2017). Therefore, the most important issue includes: how to control the pollution emissions of the MSWI process under the premise of satisfying economic benefits (Hu et al., 2018); and how to provide real-time DXN emission concentration data for the operation optimisation control of the MSWI process (Koloma et al., 2015). In the actual industrial process, the DXN emission concentration detection is mainly carried out through a combination of online sampling and offline experimental analysis. Although it can accurately measure the DXN concentration (Qiao et al., 2020), the relatively long period and high cost makes it difficult to give a direct support for the real-time optimised control of MSWI process (Zhang et al., 2008).

The complex physical and chemical characteristics of the MSWI process make it impossible to establish a DXN emission concentration mechanism model (Stanmore, 2002). Study shows that the online prediction of DXN emission is an indispensable and important issue for achieving optimal control of the MSWI process (Lavric et al., 2005). There are some researches for DXN online indirect detection (Li et al., 2015; Cao et al., 2017; Nakui et al., 2011). The first step is to detect the content of DXN-related substances with expensive complex instruments. The second step is to achieve the DXN concentration by making a mapping relation. However, it has disadvantages such as large

time lag, expensive equipment. Moreover, it is also need the offline direct analysis to construct the mapping model (Zhang et al., 2008).

The use of mechanism model or data-driven model can perform online detection of process parameters that are not easy to obtain (Tang et al., 2013). This approach has the advantages of faster and more economically than direct offline analysis and online indirect detection methods. It has widely used in actual industrial process (Souza et al., 2016). Therefore, it is necessary to construct soft measurement model to achieve real-time detection of DNX emission concentration. For the MSWI process, there have been studies using feature selection combined with neural networks to construct DXN emission predictive model (Bunsan et al., 2013; Chang and Chen, 2000; Wang et al., 2008). However, due to the inherently small samples, high dimensions, and collinearity characteristics of the modelling data, these methods has disadvantages such as easily fall into local minimum and training over fitting.

To some complex learning tasks, the performance of deep learning models is better than the traditional machine learning models. However, the learning behaviour of deep neural network (DNN) models is difficult to make analyse and interpret in theory. Moreover, it is difficult to apply to small sample data modelling. Recently, inspired by the DNN mechanism (Hinton et al., 2012), Zhou et al. proposed a deep forest (DF) algorithm for classification tasks (Zhou and Feng, 2017). It consists of multi-grained scanning and cascade forest. In nature, DF is a type of supervised non-neural network mode deep learning method based on the forest algorithm. It has good representation learning ability and can maintain good results when there are little training data. DF algorithm has been applied in many fields, such as video abnormal behaviour detection (Yang et al., 2019), flame feature extraction and flame recognition (Zhu et al., 2018), fault diagnosis of rolling bearings (Qi et al., 2018), etc. However, the above existing researches on DF is mainly used for classification problems. Therefore, the DF algorithm for the regression modelling problem has not been addressed at the present research.

Motivated by the above problem, a non-neural network mode of deep forest regression (DFR) algorithm is proposed. The proposed DFR method improves the DF structure as follows:

1 The bootstrap strategy and the random subspace method (RSM) are used to randomly sample the sample space and feature space, which replaces the multi-grained scanning module and is used to construct each layer sub-forest model.

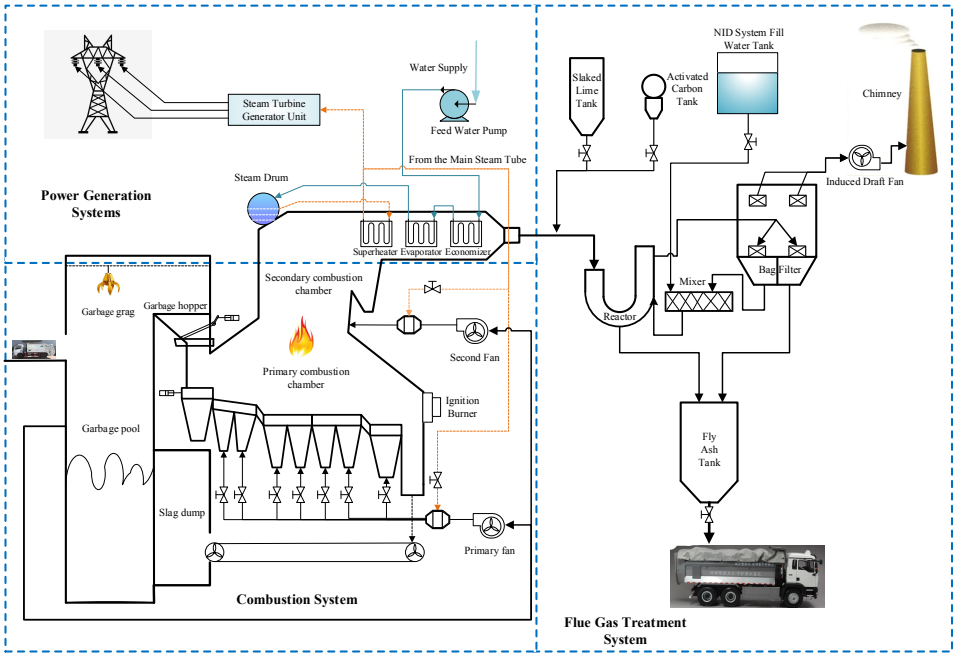
2 The DFR structure of the three-layer architecture that analogous to the DNN is proposed, and the classification tree in the cascade forest is replaced by regression tree to complete the regression modelling tasks. Moreover, the adaptive selection of the middle layer number is realised.

3 The features between two adjacent layers is represented by using the mean value of the sub-forest model prediction value vectors. This method is used to construct a soft measurement model of DXN emission concentration in the process of MSWI.

2 MSWI process and DXN emission

Figure 1 shows the process flow chart of a typical MSWI process in Beijing of China.

Figure 1 MSWI process (see online version for colours)

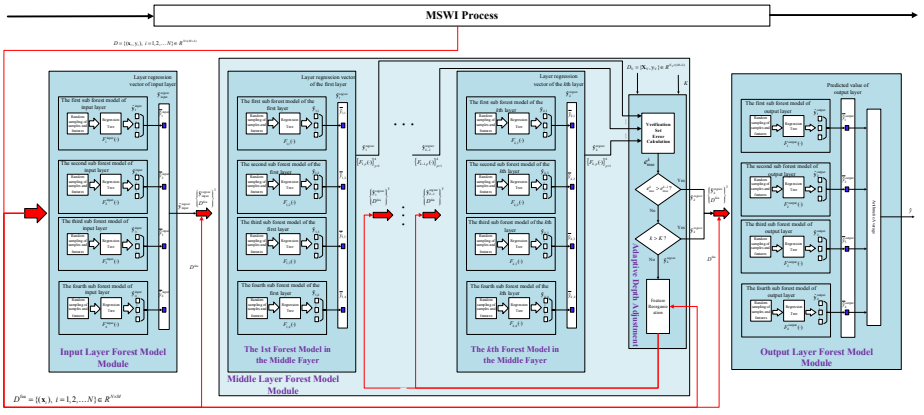


The MSWI process mainly converts MSW into residue, fly ash, flue gas and heat. In which the former three products are related to DXN emission (Mckay, 2002). There are more residues in the furnace, but the DXN concentration is lower. The amount of fly ash is less than the residue, and the DXN concentration is higher than the residue. The DXN concentration in the flue gas is between that of the residue and the fly ash, whose production includes the incomplete combustion and new synthesis reaction (Li et al., 2005). At present, for DXN detection, MSWI companies and environmental protection departments conduct offline analysis with a monthly or quarterly period. This method is not only lag-time long but also cost expensive. Thus, the DXN modelling data has characteristic of small samples number and high dimension input feature. At the same time, there are also some objective problems, such as unknown DXN content in MSW, complicated mechanism of DXN generation and absorption. Therefore, the online prediction of DXN emission concentration using soft measurement technology is according with the actual needs of the industrial process.

3 Modelling strategy

The proposed modelling strategy includes three modules, i.e., input layer forest model, middle layer forest model and output layer forest model. The structure is shown in Figure 2.

Figure 2 The proposed DRF modelling strategy $\overline{y}_{C_1}$ and $\overline{y}_{C_2}$ is the mean value of the true value in the C_1 and C_2, respectively; θ_{Forest} is the sample threshold contained in the leaf node (see online version for colours)



The description of symbols in Figure 2 is shown in Table 1.

Table 1 Description of symbols

Symbol	*Description*
D	Training set
D^{fea}	Raw feature set
D_V	Validation set
$\mathbf{x}_i$	Feature vector of training set
$\mathbf{x}_{\text{V}}$	Feature matrix of validation set
y_i	The truth value of the training set
$\mathbf{y}_{\text{V}}$	The truth vector of the validation set
$\hat{\mathbf{y}}_p^{\text{input}}$	Predicted value vector of the *p*th sub-forest model in the input layer forest model
$\overline{\hat{y}}_p^{\text{input}}$	The predicted mean of the *p*th sub-forest model in the input layer forest model
$\hat{\mathbf{y}}_{\text{input}}^{\text{regvec}}$	Layer regression vector of input layer forest model
$\hat{\mathbf{y}}_{k,p}$	The predicted value vector of the *p*th sub-forest model of the *k*th forest model in the middle layer model
$\overline{\hat{y}}_{k,p}$	The predicted mean of the *p*th sub-forest model of the *k*th forest model in the middle layer model
$\hat{\mathbf{y}}_k^{\text{regvec}}$	Layer regression vector of the *k*th forest model in the middle layer model
$\hat{\mathbf{y}}_p^{\text{output}}$	Predicted value vector of the *p*th sub-forest model in the output layer forest model
$\overline{\hat{y}}_p^{\text{output}}$	The predicted mean of the *p*th sub-forest model in the output layer forest model

Table 1 Description of symbols (continued)

Symbol	*Description*
$\hat{y}$	Predictive output of output layer forest model
$\begin{Bmatrix} \hat{\mathbf{y}}_{\text{input}}^{\text{regvec}} \\ D^{\text{fea}} \end{Bmatrix}^{\text{T}}$	Augmented layer regression vector of input layer forest model
$\begin{Bmatrix} \hat{\mathbf{y}}_{k}^{\text{regvec}} \\ D^{\text{fea}} \end{Bmatrix}^{\text{T}}$	Augmented layer regression vector of the *k*th forest model in the middle layer model
$F_p^{\text{input}}(\cdot)$	The *p*th sub-forest model in the input layer forest model
$F_{k,p}(\cdot)$	The *p*th sub-forest model of the *k*th forest model in the middle layer model
$\left\{F_{k,p}(\cdot)\right\}_{p=1}^{4}$	The *k*th forest model in the middle layer forest model
$F_p^{\text{output}}(\cdot)$	The *p*th sub-forest model in the output layer model
e_{mse}^{k}	Validation error of the *k*th forest model in the middle layer model
N	Number of samples in training set
N_V	Number of samples in validation set
M	Number of features
p	Number of sub-forests in each forest model
K	The maximum depth (layers) of the deep forest regression model

The functions are described as follows:

1 *Input layer forest model*: At first, Bootstrap and RSM methods are performed on the training set to obtain the sub-training set. Then, the input layer forest model is constructed. The predicted mean value of these sub-forest models is combined to obtain the layer regression vector. Thirdly, it is combined with the raw feature set to obtain the augmented layer regression vector, which is used as input of the middle layer forest model.

2 *Middle layer forest model*: The augmented layer regression vector of the input layer forest model is used as the input of the middle layer forest model. Then, we construct the middle layer forest model containing *k* layer forest model. By calculating the middle layer forest model's error, the depth of the forest model is adaptive adjusted with the following criterion: when the validation error no longer decreases or the depth of the middle-layer forest model reaches the pre-set value, the construction of the middle-layer forest model is stopped, the regression vector of the middle layer and the raw feature set are re-combined as the augmented layer regression vector for output layer; otherwise, continue to build the next layer of the middle layer forest model.

3 *Output layer forest model*: The augmented layer regression vector output from the middle layer forest model is used as the input to train the output layer forest model. The final prediction value is obtained by simple average of the prediction mean value of each sub-forest model.

4 Modelling methods

4.1 Input layer forest model module

At first, the construction of the *p*th sub-forest model in the input layer forest model is described. Random sampling based on bootstrap and RSM methods is made on the on training set *D*. It is shown as,

$$\{(\mathbf{x}_i^{j,M^j}, y_i^j)_p^{\text{input}}\} = f_{\text{RSM}}\left(\text{Bootstrap}(D, N), M^j\right) \tag{1}$$

where N is the number of samples in training set D; $\{(\mathbf{x}_i^{j,M^j}, y_i^j)_p^{\text{input}}\}$ is the *j*th training subset of the *p*th sub-forest; and M^j is the number of selected features, usually $M^j << M$.

Repeat the above steps to obtain *J* training subsets, and then the J regression trees of the *p*th sub-forest is constructed. We take the *j*th training subset $\{(\mathbf{x}^{j,M^j}, y^j)_n\}_{n=1}^N$ as an example to describe the construction process. The best segmentation feature and feature value is find based on the following criteria,

$$\begin{aligned}(M_{\text{sel}}^j, s) &= \min\left[\sum^{C_1}(y_{C_1}^j - \overline{y}_{C_1})^2 + \sum^{C_2}(y_{C_2}^j - \overline{y}_{C_2})^2\right] \\ s.t. &\begin{cases} C_1 > \theta_{\text{Forest}} \\ C_2 > \theta_{\text{Forest}} \end{cases}\end{aligned} \tag{2}$$

where M_{sel}^j is the best segmentation feature; s is the value of the best segmentation feature; C_1 and C_2 is the two regions into which the sample is divided by the best segmentation feature; $y_{C_1}^j$ and $y_{C_2}^j$ is the true value in the C_1 and C_1 region, respectively;

Based on the above criteria, the input feature space is divided into two regions by recursion all input features to find the optimal feature and segmentation value. Then repeat the above process for each region until the number of samples contained in the leaf node is less than threshold θ_{Forest}. Finally, the input feature space is divided into Q regions. To obtain a regression tree model, we define the following function,

$$\Gamma_{p,j}^{\text{input}}(\cdot) = \sum_{q=1}^{Q} c_{p,j}^q I(\mathbf{x}^{j,M^j} \in R_q) \tag{3}$$

where

$$c_{p,j}^q = \frac{1}{N_{R_q}} \sum_{n_{R_q}=1}^{N_{R_q}} y_{n_{Rq}}^j, \qquad N_{R_q} \le \theta_{\text{Forest}} \tag{4}$$

where N_{R_q} is the number of samples contained in the region R_q; $y^j_{n_{Rq}}$ is the n_{R_q}th measurement of the jth training subset in the R_q region; $I(\cdot)$ is an indicator function, $I(\cdot)=1$ exists at $\mathbf{x}^{j,M^j} \in R_q$, otherwise $I(\cdot)=0$ exists.

Repeat the above steps, we finally get the pth sub-forest model,

$$F_p^{\text{input}}(\cdot) = \frac{1}{J}\sum_{j=1}^{J} \Gamma_{p,j}^{\text{input}}(\cdot) \tag{5}$$

These sub-forest models are construed with RF and CRF algorithms, which are shown in Tables 2 and 3.

Table 2 RF algorithm

Algorithm 1: RF	
Input:	Training set D; number of regression trees J; number of selected features M^j; threshold of leaf node training samples θ_{RF}
Output:	RF model $F_{\text{RF}}(\cdot)$
Step 1:	Use Bootstrap and RSM to randomly sample the training set samples and features to obtain J sub-training sets
Step 2:	For 1: J
Step 3:	Using the criteria in formula (2), traverse to find the best segmentation feature number and the value of the segmentation point
Step 4:	The input feature space is divided into Q regions
Step 5:	Get the tree model $\Gamma_j^{RF}(\cdot)$
Step 6:	End for
Step 7:	Return $F_{\text{RF}}(\cdot)$

Table 3 CRF algorithm functions

Algorithm 2:CRF	
Input:	Training set D; number of regression trees J; number of selected features M^j; threshold of leaf node training samples θ_{RF}
Output:	RF model $F_{\text{RF}}(\cdot)$
Step 1:	Use Bootstrap and RSM to randomly sample the training set samples and features to obtain J sub-training sets
Step 2:	For 1: J
Step 3:	Adopt a random way to traverse to find the segmentation feature number and the value of the segmentation point
Step 4:	The input feature space is divided into Q regions
Step 5:	Get the tree model $\Gamma_j^{CRF}(\cdot)$
Step 6:	End for
Step 7:	Return $F_{\text{CRF}}(\cdot)$

Repeat the above steps, all the input layer forest models can be denoted as $\left\{F_p^{\text{input}}(\cdot)\right\}_{p=1}^{4}$.

Further, the generation process of the augmented layer regression vector of the input layer forest model is given as follows. After the J DXN emission prediction values generated by the pth sub-forest model $F_p^{\text{input}}(\cdot)$ in the input layer forest model and the prediction value vector $\hat{\mathbf{y}}_p^{\text{input}}$ is constituted, we calculate the predicted mean value $\overline{\hat{y}}_p^{\text{input}}$ of the pth sub-forest model. Thus, the layer regression vector of the input layer forest model is obtained with,

$$\hat{\mathbf{y}}_{\text{input}}^{\text{regvec}} = \{\overline{\hat{y}}_1^{\text{input}}, \ldots, \overline{\hat{y}}_4^{\text{input}}\} \tag{6}$$

At last, the raw feature set D^{fea} and the layer regression vector $\hat{\mathbf{y}}_{\text{input}}^{\text{regvec}}$ are combined in series to obtain the augmented layer regression vector $\begin{Bmatrix} \hat{\mathbf{y}}_{\text{input}}^{\text{regvec}} \\ D^{\text{fea}} \end{Bmatrix}^{\text{T}}$ of the input layer forest model. It is the input of the middle layer forest model.

4.2 The middle layer forest model module

The construction process of each layer forest model in the middle layer forest model is similar to that of the input layer forest model. The description takes the construction of the kth layer of the middle layer forest model as an example. The kth middle layer forest model training dataset D_k is the augmented layer regression vector output from the $(k-1)$th middle layer forest model,

$$D_k = \left\{ \begin{Bmatrix} \hat{\mathbf{y}}_{k-1}^{\text{regvec}} \\ D^{\text{fea}} \end{Bmatrix}^{\text{T}}, \mathbf{y} \right\} \tag{7}$$

The number of features of the training set D_k is denoted as $M_k = M + 4$. Same as that of the input layer forest, we take the pth training set of the kth middle layer forest model as an example. The generation process of the training subset can be expressed as,

$$\left\{ \{(\mathbf{x}^{j,M_k^j}, y^j)_n\}_{n=1}^{N} \right\}_{k,p} = f_{\text{RSM}}\left(\text{Bootstrap}(D_k, N), M^j\right) \tag{8}$$

where $\left\{ \{(\mathbf{x}^{j,M_k^j}, y^j)_n\}_{n=1}^{N} \right\}_{k,p}$ is the jth training subset of the pth sub-forest model in the kth layer forest model; and M_k^j is the number of selected features, usually $M_k^j << M_k$. We repeat the above steps to obtain J training subset, and then J regression trees in the pth sub-forest in the kth middle layer forest model is constructed. Here, we take the jth training subset $\left\{ \{(\mathbf{x}^{j,M_k^j}, y^j)_n\}_{n=1}^{N} \right\}_{k,p}$ as an example. Based on the criterion of equation (2), the following equation is used to obtain the regression tree model,

$$\Gamma_{p,j}^{k}(\cdot)=\sum_{q=1}^{Q}c_{p,j}^{q}I(\mathbf{x}^{j,M^{j}}\in R_{q}) \tag{9}$$

where

$$c_{p,j}^{q}=\frac{1}{N_{R_q}}\sum_{n_{R_q}=1}^{N_{R_q}}y_{n_{R_q}}^{j},\qquad N_{R_q}\le\theta_{\text{Forest}} \tag{10}$$

Repeat the above steps to get the sub-forest model of the kth middle layer forest model,

$$F_{k,p}(\cdot)=\frac{1}{J}\sum_{j=1}^{J}\Gamma_{p,j}^{k}(\cdot) \tag{11}$$

Same as that of the input layer forest, all the kth layer middle-layer forest models constructed with RF and CRF algorithms can be denoted as $\left\{F_{k,p}(\cdot)\right\}_{p=1}^{4}$.

Further, the layer regression vector of the kth forest model is denoted as,

$$\hat{\mathbf{y}}_{k}^{\text{regvec}}=[\bar{\hat{y}}_{k,1},\ldots,\bar{\hat{y}}_{k,4}] \tag{12}$$

Then, the augmented layer regression vector of the middle layer forest model is denoted as $\begin{Bmatrix}\hat{\mathbf{y}}_{k}^{\text{regvec}}\\ D^{\text{fea}}\end{Bmatrix}^{\text{T}}$. However, the adaptive depth adjustment of the middle layer number have to be addressed for different modelling data.

Here, the validation set of the DFR model is denoted as $D_V=\{\mathbf{X}_{\text{V}},\mathbf{y}_{\text{V}}\}\in R^{N_{\text{V}}\times(M+1)}$. The prediction value of the kth forest model in the middle-layer forest model is calculated with,

$$\hat{y}_{k}^{i}=\frac{1}{4}\sum_{p=1}^{4}F_{k,p}(\cdot) \tag{13}$$

where $\hat{y}_{k}^{i}$ is the prediction value of the kth forest model for the ith sample in the validation set; and N_{V} is the number of validation set samples.

We calculate the prediction error of the kth layer forest model as,

$$e_{\text{rmse}}^{k}=\frac{1}{N_{\text{V}}}\sum_{i=1}^{N_{\text{V}}}(\hat{y}_{k}^{i}-y_{\text{V}}^{i})^{2} \tag{14}$$

where y_{V}^{i} is the true value in the validation set.

Next, we compare e_{rmse}^{k} and e_{rmse}^{k-1}:

1 If $e_{\text{rmse}}^{k}<e_{\text{rmse}}^{k-1}$ and $k\le(K-2)$, the augmented layer regression vector of the kth forest model is obtained by using the layer regression vector $\hat{\mathbf{y}}_{k}^{\text{regvec}}\in R^{N\times4}$ of the kth forest model and the raw feature set D^{fea}. Then, the next layer forest model is be trained continually;

2 If $e_{\text{rmse}}^{k} \geq e_{\text{rmse}}^{k-1}$ or $k > (K-2)$, the number of the middle layer forest model is satisfied. The augmented layer regression vector $\begin{Bmatrix} \hat{\mathbf{y}}_k^{\text{regvec}} \\ D^{\text{fea}} \end{Bmatrix}^{\text{T}}$ is used as the output of the middle layer forest model.

4.3 Output layer forest model module

The training set $D_{\text{output}} = \{(\mathbf{x}_{\text{output},i}, \text{y}_i)\}$ of the output layer forest model can be denoted as,

$$D_{\text{output}} = \left\{ \begin{Bmatrix} \hat{\mathbf{y}}_k^{\text{regvec}} \\ D^{\text{fea}} \end{Bmatrix}^{\text{T}}, \mathbf{y} \right\} \tag{15}$$

Same as that of the input layer forest, Bootstrap and RSM methods are used again to generate the training subset. Taking the pth training set as an example, it is shown as,

$$\left\{ \{(\mathbf{x}^{j,M_k^j}, y^j)_n\}_{n=1}^{N} \right\}_{\text{output},p} = f_{\text{RSM}}\left(\text{Bootstrap}(D_{\text{output}}, N), M^j\right) \tag{16}$$

where $\left\{ \{(\mathbf{x}^{j,M_k^j}, y^j)_n\}_{n=1}^{N} \right\}_{\text{output},p}$ is the jth training subset of the pth sub-forest model in the output layer forest model.

Further, based on the criterion of equation (2), the following equation is used to obtain the regression tree model,

$$\Gamma_{p,j}^{\text{output}}(\cdot) = \sum_{q=1}^{Q} c_{p,j}^{q} I(\mathbf{x}^{j,M^j} \in R_q) \tag{17}$$

where

$$c_{p,j}^{q} = \frac{1}{N_{R_q}} \sum_{n_{R_q}=1}^{N_{R_q}} y_{n_{R_q}}^{j}, \qquad N_{R_q} \leq \theta_{\text{Forest}} \tag{18}$$

Repeat the above steps to get the pth sub-forest model of the output layer forest model.

$$F_p^{\text{output}}(\cdot) = \frac{1}{J} \sum_{j=1}^{J} \Gamma_{p,j}^{\text{output}}(\cdot) \tag{19}$$

Same as that of the input layer forest, all the output layer forest models constructed with RF and CRF algorithms can be denoted as $\left\{F_p^{\text{output}}(\cdot)\right\}_{p=1}^{4}$. Based on the J prediction values generated, a prediction value vector $\hat{\mathbf{y}}_p^{\text{output}}$ is formed, and the prediction mean $\overline{\hat{y}}_p^{\text{output}}$ of the pth sub-forest model in the output layer is calculated. Repeat the above steps to get the predicted output set $\left\{\overline{\hat{y}}_p^{\text{output}}\right\}_{p=1}^{4}$ of the output layer forest model. Finally, the

predicted values of the output layer sub-forest model is calculated by simple average method,

$$\hat{y} = \frac{1}{4}\sum_{p=1}^{4} \overline{\hat{y}}_p^{\text{output}} \tag{20}$$

where $\hat{y}$ is the final predicted output.

4.4 Algorithm pseudo code

The pseudo code of the proposed DFR algorithm is shown in Table 4.

Table 4 Pseudo code of DFR algorithm

Algorithm 3:DFR	
Input:	Training set $D = \{(\mathbf{x}_i, \mathrm{y}_i)\} \in R^{N\times(M+1)}$; Validation set $D_V = \{\mathbf{X}_\mathrm{V}, \mathbf{y}_\mathrm{V}\}$; Number of features selected M^j; Number of regression trees *J*; Leaf node training sample threshold θ_{Forest};Maximum number of layers *K*
Output:	Predictive value $\hat{y}$
Step 1:	For 1: $p/2$
Step 2:	RF(D, J, M^j, θ_{Forest})
Step 3:	Calculate the predicted value of RF and put it into the layer regression vector $\hat{\mathbf{y}}_{\text{input}}^{\text{regvec}}$
Step 4:	CRF (D, J, M^j, θ_{Forest})
Step 5:	Calculate the predicted value of CRF and put it into the layer regression vector $\hat{\mathbf{y}}_{\text{input}}^{\text{regvec}}$
Step 6:	End for
Step 7:	Combine $\hat{\mathbf{y}}_{\text{input}}^{\text{regvec}}$ with the original feature set D^{fea} to get the augmented layer regression vector $\begin{Bmatrix}\hat{\mathbf{y}}_{\text{input}}^{\text{regvec}} \\ D^{\text{fea}}\end{Bmatrix}^{\mathrm{T}}$ of the input layer forest model
Step 8:	For 1: *k*
Step 9:	Combine the augmented layer regression vector $\begin{Bmatrix}\hat{\mathbf{y}}_{k-1}^{\text{regvec}} \\ D^{\text{fea}}\end{Bmatrix}^{\mathrm{T}}$ of the previous model (the augmented layer regression vector $\begin{Bmatrix}\hat{\mathbf{y}}_{\text{input}}^{\text{regvec}} \\ D^{\text{fea}}\end{Bmatrix}^{\mathrm{T}}$ of the input layer forest model in the first loop) with the DXN truth value of the training set to obtain the *k*th forest model of the middle layer forest model. Input training set $D_k = \{(\mathbf{x}_{k,i}, \mathrm{y}_i)\} \in R^{N\times(M_k+1)}$
*Step*10:	For 1:*p*
*Step*11:	RF(D_k, J, M^j, θ_{Forest})
Step 12:	Calculate the predicted value of RF and put it into the layer regression vector $\hat{\mathbf{y}}_k^{\text{regvec}}$

Table 4 Pseudo code of DFR algorithm (continued)

Algorithm 3:DFR	
Step 13:	CRF(D_k, J, M^j, θ_{Forest})
Step 14:	Calculate the predicted value of CRF and put it into the layer regression vector $\hat{\mathbf{y}}_k^{\text{regvec}}$
Step 15:	End for
Step 16:	Combine $\hat{\mathbf{y}}_k^{\text{regvec}}$ with the original feature set D^{fea} to get the augmented layer regression vector $\begin{Bmatrix} \hat{\mathbf{y}}_k^{\text{regvec}} \\ D^{\text{fea}} \end{Bmatrix}^{\text{T}}$ of the kth forest model in the middle-layer forest model
Step 17:	Calculate the prediction error e_{rmse}^k of the current layer forest model in the validation set
Step 18:	IF $e_{\text{rmse}}^k < e_{\text{rmse}}^{k-1}$ and $k \le (K-2)$; Continue to build the next forest model of the middle forest model
Step 19:	IF $e_{\text{rmse}}^k \ge e_{\text{rmse}}^{k-1}$ and $k > (K-2)$; Stop building the middle layer forest model and use the augmented layer regression vector $\begin{Bmatrix} \hat{\mathbf{y}}_k^{\text{regvec}} \\ D^{\text{fea}} \end{Bmatrix}^{\text{T}}$ as the output of the middle forest model
Step 20:	End for
Step 21:	Combine the augmented layer regression vector output from the middle layer forest model with the DXN truth value in the training set to obtain the output layer forest model input training set $D_{\text{output}} = \{(\mathbf{x}_{\text{output},i}, \mathrm{y}_i)\} \in R^{N\times(M_{\text{output}}+1)}$
Step 22:	For 1: p
Step 23:	RF(D_{output},J, M^j, θ_{Forest})
Step 24:	Calculate the predicted mean $\overline{\hat{y}}_p^{\text{output}}$ of the pth sub-forest model in the output layer
Step 25:	CRF(D_{output},J, M^j, θ_{Forest})
Step 26:	Calculate the predicted mean $\overline{\hat{y}}_p^{\text{output}}$ of the pth sub-forest model in the output layer
Step 27:	End for
Step 28:	Obtain the predicted output set $\left\{\overline{\hat{y}}_p^{\text{output}}\right\}_{p=1}^{4}$ of the output layer forest model, arithmetically average $\left\{\overline{\hat{y}}_p^{\text{output}}\right\}_{p=1}^{4}$ to obtain the final DXN predicted value $\hat{y}$

5 Simulation results

5.1 Concrete compressive strength data

5.1.1 Modelling data

The concrete compressive strength data set provided by the University of California Irvine (UCI) platform (Yeh, 1998; Tang et al., 2012) was used to verify the DFR method. The data set contains 1030 samples and 8 features. In this paper, 1/2 of 1030 samples are selected as training samples, 1/4 as validation samples, and 1/4 samples as test samples.

5.1.2 Modelling results

The number of input features and regression trees are selected as M^j=4 and $J=500$ according to prior knowledge. Based on the number of training samples, the threshold interval of the leaf nodes is selected as [5, 50]. By set the maximum number of layers $K=50$, the relationship between the leaf nodes samples threshold (MinSamples) and the validation error is shown in Figure 3.

Figure 3 shows that the validation error has the minimum RMSE with leaf nodes samples threshold value θ_{Forest}=10. Therefore, we set θ_{Forest}=10, $J=500, K=50$. The relationship between the number of input features and the validation error is shown in Figure 4.

Figure 3 Validation error under different leaf nodes samples thresholds (see online version for colours)

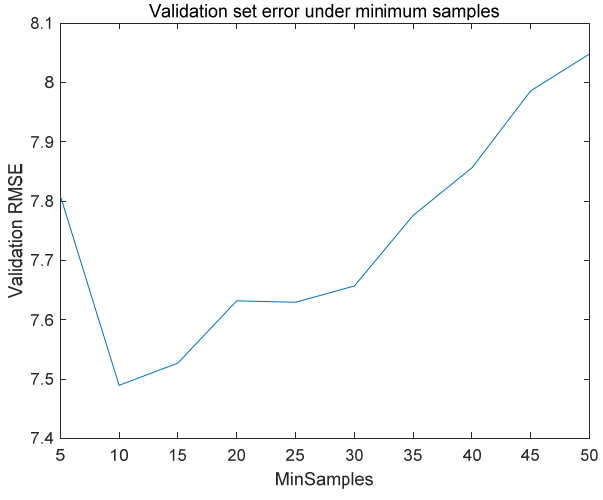


Figure 4 Validation error under different feature numbers (see online version for colours)

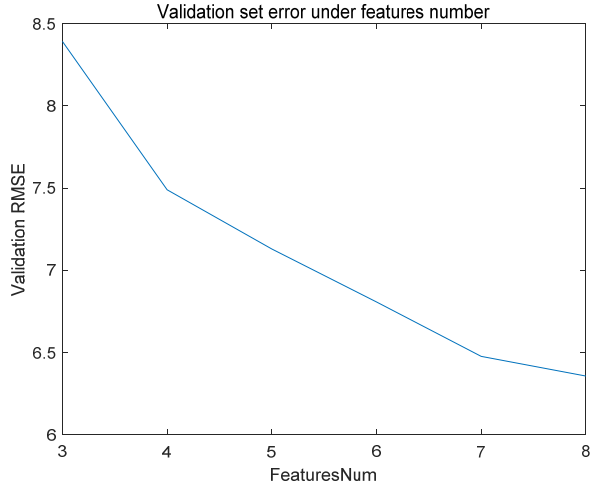


Figure 4 shows that the validation error decreases with the number of input features increase. When M^j=8, the validation error reaches the minimum value. Therefore, we set θ_{Forest}=10, M^j=8, $K = 50$. The relationship between the validation error and the regression trees number J is shown in Figure 5.

Figure 5 Validation error under different regression tree numbers (see online version for colours)

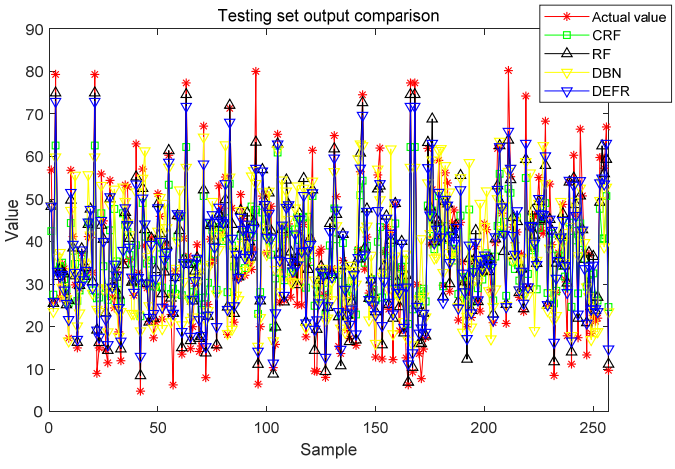
Figure 5 shows that the validation error reaches the minimum with $J = 50$.

Based on the above experimental analysis, the parameters of the DFR model are set as: $K = 50, \theta_{\text{Forest}} = 10, M^j = 8, J = 50$.

5.1.3 Comparative results

The testing curves of different methods are shown in Figure 6.

Figure 6 Testing curves of concrete compressive strength data (see online version for colours)



The statistical results of different modelling methods are shown in Table 5.

Table 5 Statistical results of different methods

	RMSE			*Remark*
	Training	*Validation*	*Testing*	
CRF	8.8194	10.4088	10.2828	
RF	3.9796	6.4405	6.0188	
DBN	10.1151	10.3689	11.3083	
DFR	3.1368	6.3315	5.9825	Layer = 3

Figure 6 and Table 5 show that:

1 DBN has the largest prediction error on all data sets

2 RF has better prediction performance than CRF duo to its minimum average error rule

3 the proposed DFR method has the best prediction performance in all data set.

However, the super parameters of DRF are not be optimised jointly. Thus, the prediction performance can be improved further. More researches should be done in the future study.

5.2 DXN emission data of the MSWI process

5.2.1 Modelling data

The data come from the 1# and 2# furnaces of an actual MSWI process in Beijing of the past 6 years. The number of the samples and input feature are 67 and 287, respectively. In which, 1/2 of all 67 samples are used as training data, 1/4 is used as validation data, and 1/4 is used as testing data.

5.2.2 Modelling results

According to empirical knowledge, the number of input features and regression trees are set as M^j=17 and $J=500$. By set the maximum number of layers $K=50$, the relationship between the leaf node sample threshold θ_{Forest} and the validation error is shown in Figure 7.

Figure 7 shows that the validation error has the minimum value with θ_{Forest}=4. Further, by set θ_{Forest}=4, $J=500, K=50$, the relationship between the input features number M^j and the validation error is shown in Figure 8.

Figure 8 shows that the validation error reaches the minimum with the features number M^j=47. As the number of features increases, the validation error shows a significant downward trend. It shows that the modelling parameters can be further optimised.

Then, we set θ_{Forest}=4, M^j=47, $K=50$. The relationship between the regression trees number J and the validation error is shown in Figure 9.

Figure 7 Validation error under different leaf node sample thresholds (see online version for colours)

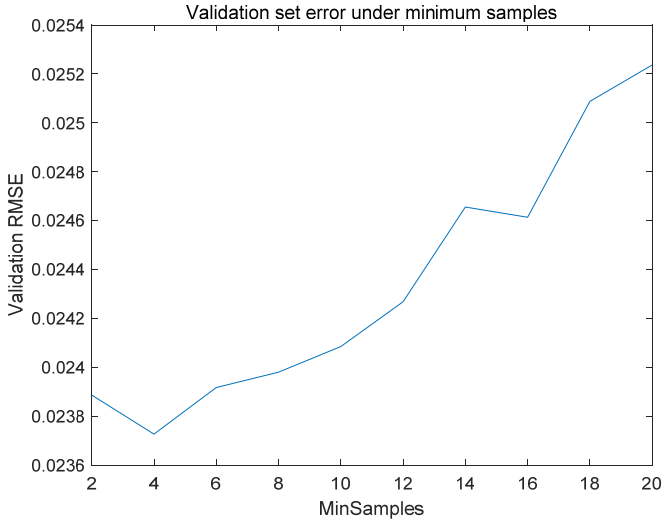


Figure 8 Validation error under different feature number (see online version for colours)

Figure 9 shows that the validation error reaches the minimum value with $J = 350$.

Based on the above experimental analysis, the parameters of the DXN soft sensor model are set as $K = 50, \theta_{\text{Forest}} = 4, M^j = 47, J = 350$.

5.2.3 Comparative results

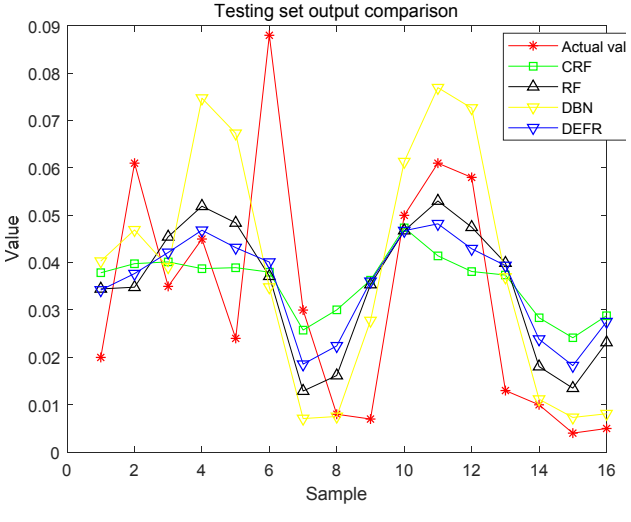
The statistical results of different modelling methods are shown in Table 6. The prediction curves and RMSE statistical results are shown in Figure 10 and Table 2, respectively.

Figure 9 Validation error under different numbers of regression trees (see online version for colours)

Table 6 Statistical results of different methods

	RMSE			*Remark*
	Train	*Validation*	*Test*	
CRF	0.0167	0.0260	0.0218	
RF	0.0087	0.0205	0.0206	
DBN	0.0047	0.0158	0.0229	
DFR	0.0116	0.0228	0.0203	Layer = 3

Figure 10 DXN testing data prediction curve (see online version for colours)



It can be seen from Table 6:

1 DBN has the smallest prediction error in the training set and validation set of the DXN high-dimensional small sample data set, but its prediction performant in the test set is the worst. Thus, over fitting is one of the main reason;

2 The DFR method using CRF and RF as the learner has the highest prediction accuracy for the DXN emission concentration in the testing set. It obtains the trade-off between the training and testing accuracy. Thus, combined with the simulation experiment of the previous data set, it shows that DRF has better generalisation performance and stability than DBN approach;

3 The results of CRF and RF model shows that different base learners have different prediction performance. Future research should address on how to choose more complementary base learners to improve diversity among different layer's sub-models.

6 Conclusions

A soft measurement method of DXN emission concentration based on non-neural network mode deep learning regression forest algorithm is proposed. The innovation lies in proposing a new DFR algorithm for regression problem modelling. The detailed contribution include:

1 the classification tree of original DF algorithm is modified into a regression tree

2 the predicted values of the sub-forest models are used to augment feature representation among different layers

3 at the same time, the validation error is added to the level of the middle layer forest model for realising the adaptive adjustment of the DFR model's depth.

Finally, the effectiveness of the proposed method was verified by using UCI benchmark data and actual industrial DXN data. One of the next research direction is how to make optimised selection of the super parameters of the proposed DRF method.

Acknowledgements

Fund projects: National Natural Science Foundation of China (62073006, 62021003, 61890930-5, 61873009), Beijing Natural Science Foundation (4212032, 4192009), National Key R&D Program of the Ministry of Science and Technology (2018YFC1900800-5).

References

Bunsan, S., Chen, W.Y., Chen, H.W., Chuang, Y.H. and Grisdanurak, N. (2013) 'Modeling the dioxin emission of a municipal solid waste incinerator using neural networks', *Chemosphere*, Vol. 92, pp.258–264.

Cao, Y., Shang, F-J. and Pan, D-G. (2017) *Gas Chromatography-Mass Spectrometry Transmission Line System for On-Line Detection of Dioxins*, China, CN206378474U, 2017-08-04.

Chang, N.B. and Chen, W.C. (2000) 'Prediction of PCDDs/PCDFs emissions from municipal incinerators by genetic programming and neural network modeling', *Waste Management and Research*, Vol. 18, pp.41–351.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mhamed, A., Jaitly, N., Senior, Andrew., Vanhoucke, V., Nguyen, P., Sainath, T. and Kingsbury, B. (2012) 'Deep neural networks for acoustic modeling in speech recognition', *IEEE Signal Processing Magazine*, Vol. 29, No. 6, pp.82–97.

Hu, Y.A., Cheng, H.F. and Tao, S. (2018) 'The growing importance of waste-to-energy (WTE) incineration in China's anthropogenic mercury emissions: Emission inventories and reduction strategies', *Renewable and Sustainable Energy Reviews*, Vol. 97, pp.119–137.

Koloma, H., Takeuchi, S., Iida, M., Nakayama, S.F. and Shiozaki, T. (2015) 'A sensitive, rapid, and simple DR-EcoScreen bioassay for the determination of PCDD/Fs and dioxin-like PCBs in environmental and food samples', *Environmental Science and Pollution Research*, Vol. 22, pp.1–12.

Lavric, E.D., Konnov, A.A. and Ruyck, J.D. (2005) 'Surrogate compounds for dioxins in incineration. A review', *Waste Management*, Vol. 25, No. 7, pp.755–765.

Li, A.D., Hong, W. and Wang, J. (2015) 'Online detection of dioxin and dioxin-related substances using laser desoption/ laser ionization-mass spectrometry', *Journal of Yanshan University*, Vol. 39, No. 6, pp.511–515.

Li, H.Y., Zhang, S.T. and Zhao, X.H. (2005) 'Detection methods of dioxins emitted from municipal solid waste incinerator', *Journal of Fuel Chemistry and Technology*, Vol. 33, No. 3, pp.379–384.

Li, X.M., Zhang, C.M., Li, Y.Z. and Zhia, Q. (2016) 'The status of municipal solid waste incineration (MSWI) in China and its clean development', *Energy Procedia*, Vol. 104, pp.498–503.

Lu, J.W., Zhang, S., Hai, J. and Lei, M. (2017) 'Status and perspectives of municipal solid waste incineration in China: a comparison with developed regions', *Waste Management*, Vol. 69, pp.170–186.

Mckay, G. (2002) 'Dioxin characterisation, formation and minimisation during municipal solid waste (MSW) incineration: review', *Chemical Engineering Journal*, Vol. 86, No. 3, pp.343–368.

Nakui, H., Koyama, H., Takakura, A. and Watanabe, N. (2011) 'Online measurements of low-volatile organic chlorine for dioxin monitoring at municipal waste incinerators', *Chemosphere*, Vol. 85, No. 2, pp.151–155.

Qi, L., Gao, H., You, Z., Song, H. and Li, Z. (2018) 'Gcforest-based fault diagnosis method for rolling bearing', *Prognostics and System Health Management Conference (PHM-Chongqing)*, Chongqing, pp.572–577.

Qiao, J.F., Guo, Z.H. and Tang, J. (2020) 'Summarization of dioxin emission concentration detection methods for urban solid waste incineration process', *Acta Automatica Sinica*, Vol. 46, No. 06, pp.1063–1089.

Souza, F.A.A., Araújo, R. and Mendes, J. (2016) 'Review of soft sensor methods for regression applications', *Chemometr. Intell. Lab. Syst*., Vol. 152, pp.69–79.

Stanmore, B.R. (2002) 'Modeling the formation of PCDD/F in solid waste incinerators', *Chemosphere*, Vol. 47, pp.565–773.

Tang, J., Chai, T.Y., Yu, W. and Zhao, L.J. (2013) 'Modeling load parameters of ball mill in grinding process based on selective ensemble multisensor information', *IEEE Transactions on Automation Science and Engineering*, Vol. 10, pp.726–740.

Tang, J., Yu, W., Chai, T.Y. and Zhao, L.J. (2012) 'On-line principal component analysis with application to process modeling', *Neurocomputing*, Vol. 82, No. 1, pp.167–178.

Wang, H.R., Zhang, Y. and Wang, H. (2008) 'A study of GA-BP based prediction model of dioxin emission from MSW incinerator', *Microcomputer Information*, Vol. 24, No. 21, pp.222–224.

Yang, X.X., Li, H.B. and Hu, G. (2019) 'An anomalous behavior detection algorithm based on unbalanced deep forests', *Journal of the Chinese Academy of Electronics Sciences*, Vol. 2019, No. 9, pp.935–942.

Yeh, I.C. (1998) 'Modeling of strength of high performance concrete using artificial neural networks', *Cement and Concrete Research*, Vol. 28, No. 12, pp.1797–1808.

Zhang, H.J., Ni, Y.W., Chen, J.P. and Zhang, Q. (2008) 'Influence of variation in the operating conditions on PCDD/F distribution in a full-scale MSW incinerator', *Chemosphere*, Vol. 70, No. 4, pp.721–730.

Zhang, Y.Y., Jing, Y.S., Gao, Q.X. and Ma, Z.Y. (2011) 'Disposal situation of municipal solid waste in China and inspiration for GHG reduction', *Environmental Science Research*, Vol. 24, No. 8, pp.909–916.

Zhou, Z.H. and Feng, J. (2017) *Deep Forest: Towards an Alternative to Deep Neural Networks*, eprintarXiv: 1702.08835.

Zhu, X.Y., Yan, Y.Y., Liu, Y. and Gao, S.B. (2018) 'Flame detection based on deep forest model', *Computer Engineering*, Vol. 44, No. 7, pp.264–270.