

Multi-objective Genetic Algorithm Based Selective Neural Networks Ensemble for Concentration Estimation of Indoor Air Pollutants Using Electronic Nose

Chaibou Kadri^{1*}, Fengchun Tian¹, Lei Zhang¹, Xiongwei Peng¹, and Xin Yin¹

¹College of Communication Engineering, Chongqing University
ShaZheng street 174, ShaPingBa district, Chongqing 400044, China

Abstract

Neural networks ensemble or committee of neural networks is a learning approach where many neural networks are combined to solve a given problem. This approach has been proved to improve the generalization performance of individual networks (base networks), provided these networks are accurate enough while being error-independent (diverse). In this paper, variance inflation factor (VIF) is defined as diversity measure. A multi-objective genetic algorithm (MOGA) with two objectives (ensemble error and the new diversity metric) is used to select appropriate members of the ensemble from a pool of trained neural networks. The proposed method herein called MOGASEN (Multi Objective Genetic Algorithm based Selective ensemble) and other popular ensemble approaches were evaluated on data from an electronic nose (E-Nose) for concentration estimation of four indoor air pollutants (formaldehyde, benzene, toluene, and carbon monoxide). Empirical results show that the proposed method, while having higher capability in reducing the size of the ensemble, was, in most cases, able to outperform other methods.

Keywords: *Neural network ensemble, Electronic nose, variance inflation factor, Multi-objective genetic algorithm, air quality monitoring*

1. Introduction

Neural network ensemble (NNE) is a learning method where multiple neural networks are trained to fulfill a given task, and their predictions are combined to form the ensemble's output [1]. Since its inception, this approach to learning has been successfully applied in many domains, including medical diagnosis [2], electronic nose systems [3, 4], optical character recognition [5], and so forth.

There are generally two steps in the course of construction of neural network ensemble: training of the base networks, and combination of the trained networks. During the training phase, the main goal is to obtain networks with acceptable accuracy while committing their errors differently. The last criterion is commonly known as diversity (either implicit or explicit diversity).

Bagging and boosting, which operate by changing the training data, are the most widely used methods to generate diverse base networks. Bagging is a name derived from bootstrap aggregation; it is an effective method of ensemble learning introduced by Breiman [6]. The method uses bootstrap sampling to generate multiple data sets from the original training data, and then each of these data sets is used to train a specific model. The output of the ensemble is obtained by averaging the outputs of all the models (for regression) or through voting (for classification). It is worth noting that bagging is more effective on unstable (i.e. a small change in the training set can cause a significant change in the model) models [6] such as neural networks, regression trees, etc... Moreover, Opitz and Maclin [7] compared bagging and two boosting methods: they concluded that, as a general method, bagging is the most appropriate. As a result of that, bagging is considered in this paper. It is worth mentioning that in the literature, other methods which operate differently from bagging and boosting are also reported, with some as follows. Krogh and Vedelsby [8] use cross-validation technique to generate several base networks. Opitz and Shavlik [9] use genetic algorithm with accuracy and ambiguity as search criterion (fitness) to generate diverse and accurate base networks for classification. In [10], Yao and Liu evolve a population of neural networks and consider the individuals in the last generation as base networks. In [11], Zhi-Hua Zhou et al. employ an approach named GASEN which first trains the base networks using bootstrap replicates of the original training data, after that it assigns random weights to those base networks and uses GA to evolve the weights. At the end, base networks with weights above a designed threshold are selected to form the ensemble. Empirical results show that this method compares favorably with other popular ensemble methods.

Having a given number of base networks at hand, the next step is to combine them. The most widely used methods for this task are simple averaging or weighted sum [12, 13] for regression problems, and voting for classification problems [1]. Other methods for combining base networks are also reported in the literature [14, 15, 16]. In this paper, as the

problem at hand is a regression task and the focus is not on combination method, the output of the ensemble is obtained by averaging (simple) the predictions of the selected base networks.

It is worth mentioning that, although the general practice in most ensemble methods is to consider all base networks as component networks of the ensemble, Zhi-Hua Zhou et al. [11] have demonstrated the benefit of considering several base networks instead of all. In this paper, a similar approach is adopted. However, to select the most effective base networks, a new method is proposed. The method uses a multi-objective genetic algorithm (GA) with two objectives: ensemble error and the diversity metric (VIF), to select appropriate members of the ensemble from a pool of trained neural networks. Classical method of combining multiple objectives generally requires normalization of the individual objectives to get the final objective [9]. This encourages us to use MOGA, although, at the last generation, it requires selecting one solution from the Pareto-optimal front. Indeed all the solutions are optimal. The proposed method and some other approaches were evaluated on data from an electronic nose (E-Nose) for concentration estimation of four indoor air pollutants (formaldehyde, benzene, toluene, and carbon monoxide). An electronic nose is an artificial olfaction system which uses a finite number of partially selective sensors along with associated circuitry and a suitable signal processing system. Electronic nose systems find application in many fields which include industrial hazards monitoring, homeland security, food quality, public health, and environmental pollution. Owing to their versatility and ease of use, these systems can be a better alternative to conventional methods (gas chromatography, mass spectrometry) for continuous real-time monitoring and control of indoor air quality. However, their performance depends on the calibration model generally built using some prior measurements. This is the rationale behind using data from such an important instrument to evaluate the methods considered in this paper.

2. Experimental Details

2.1. Data sets generation

Our E-nose consists of eight sensors: two auxiliary sensors (temperature and humidity module), and six gas sensors (GSBT11, TGS2620, TGS2602, dual sensor TGS2201 with two outputs named TGS2201A and TGS2201B, and one O₂ sensor). These sensors are mounted on a self-made printed circuit board (PCB), along with associated circuitry. An analog-digital converter (AD) is used as interface between the FPGA processor and the sensors. Also, an additional flash memory is used for real-time data storage.

The sampling rate during data acquisition was one point every three seconds. For further processing, the saved data can be transferred to a personal computer (PC) using Nios II IDE and the Joint Test Action Group (JTAG) cable. Figure 1 shows our electronic nose system.



Fig. 1. Photograph of the implemented E-nose (left), and inside components (right),

All experiments were carried out in an atmosphere-controlled chamber by exposing our E-nose to four gas analytes each at different concentrations. Detailed description of the experimental setup and procedure can be found in our previous publications [17,18]. However, to make the paper self-contained, we reproduce the experimental setup (see Figure 2). As for the experimental procedure it is worth noting that during all the experiments, the respective ranges of the temperature and humidity were 15-45°C and 25-80%. Also, a single experiment consists of three phases: exposure to clean air for 120s to stabilize the sensors, exposure to gas analyte for 480s, and another exposure to clean air for 120s to allow the sensors recover. Between any two consecutive experiments, the chamber is cleaned for about 10mins to avoid (minimize) interference from any chemical remnant. It is worth mentioning that the real concentration of benzene was determined by gas chromatography method; while that of formaldehyde was determined using two different methods: acetylacetone spectrophotometric method for concentrations greater than 0.5ppm, and the 3-methyl-2-benzothiazolinone hydrochloride (MBTH) method for concentrations less than 0.5ppm. This is the aim of using organic gas sampler. For the other gases, standard measurement equipments were placed inside the chamber and displayed concentrations were recorded. The number of measurements for formaldehyde, benzene, toluene, and carbon monoxide is 126, 72, 66, and 58, at concentration ranges of 0.04-6ppm, 0.17-1ppm, 0.04-0.15ppm, and 4-55ppm, respectively.

Thus, for each gas analyte, an original data set is obtained, which contains raw measurements from the sensor array.

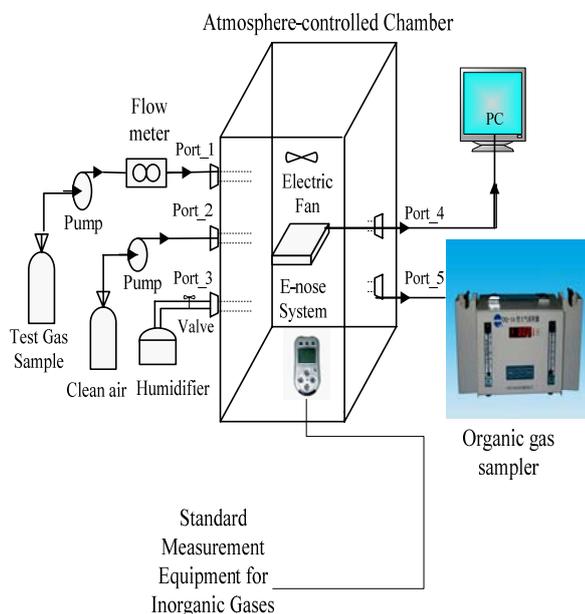


Fig. 2. Experimental setup

Prior to feature extraction, raw measurements are filtered to remove measurement noise. For gas concentration estimation, Szczurek et al. [19] demonstrated that features from the steady-state portion of a gas sensor response are more informative. Taking this into account, we selected one feature from that portion (see Figure 3). For the auxiliary sensors (temperature, humidity) we selected features at the same time positions with other sensors. The extracted features are normalized to have values in the interval [0, 1]. Having an array of eight sensors, an $8 \times m$ (m is the number of measurements or samples) feature data matrix is formed for each data set. Then we used Kennard and Stone (K-S) algorithm [20] to divide each data set into three sub data sets: 50% for training, 25% for validation, and 25% for test.

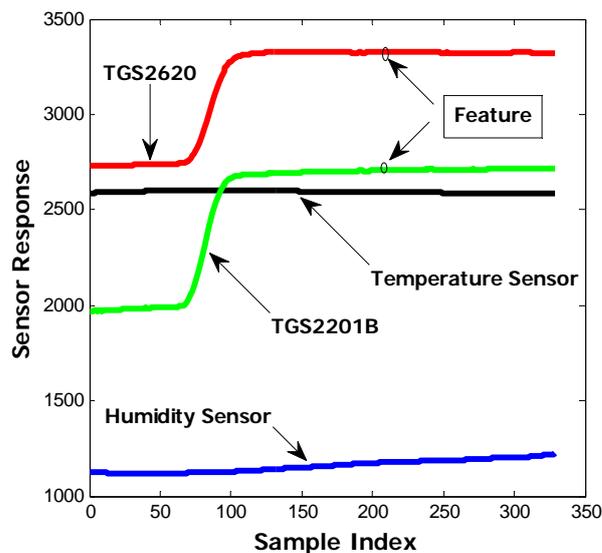


Fig. 3. Responses of two gas sensors when exposed to 5.30ppm of formaldehyde, at 45°C, 50% RH (humidity and temperature sensors' responses are shown for interpretation)

2.2 Multi-objective genetic algorithm based selective ensemble

Evolutionary techniques can be set to optimize single objectives or multiple objectives. The goal of multi-objective optimization (MOO) is to find solutions that are optimal, or at least acceptable, according to all criteria simultaneously. The most primitive form of MOO is to combine multiple objectives into a scalar fitness function. And the simplest form of this combination is a (scaled) linear combination of the different objectives.

Another sound alternative to the approach mentioned above is to keep the objectives apart. In fact, the main motivation for keeping the objectives apart is to encourage diversity among solutions, which encourages us to adopt the alternative. It is worth mentioning that a key idea in MOO is the notion of *Pareto dominance*. For instance, given a set of solutions S , a solution a_i is non-dominated if and only if there is no other alternative $a_j \in S$, $j \neq i$ so that a_j is better than a_i on all criteria. Or, expressing the opposite relation less formally, a solution is said to Pareto dominate another solution if it is as good as that solution on all objectives and better on at least one objective. This results in a partial ordering, where several solutions can be non-dominated, and thus constitute the set of best solutions for the particular set of objectives. The set of all non-dominated solutions in the search space are called the *Pareto front*, or the *Pareto optimal set*.

In [6], it has been pointed out that implicit diversity can be achieved through bagging. However, in practical applications where limited number of samples is available, this diversity is not guaranteed. We therefore used multi-objective genetic algorithm (MOGA) to further optimize the derived ensemble. More specifically, the main idea behind our approach is to consider many networks trained using bagging algorithm and keep a subset of the networks that are both accurate and diverse. Genetic algorithms are effective in their use of global information [21]; they allow us to consider a wide variety of networks during our search, so they are suitable for our search method. The Multi-Objective Genetic Algorithm function ‘*gamultiobj*’ in MATLAB was used. Each gene in the GA is a bit string of length L (the number of ANN generated using bagging), where a ‘1’ in any location indicates that the ANN with corresponding index should be included in the ensemble.

2.2.1 Variance inflation factor as diversity metric

In the course of ensemble construction diversity is one of the most important criteria. However, most of the diversity metrics are directly applicable to classification ensembles rather than regression ensembles. In this paper, we explore the possibility of using variance inflation factor (VIF) as diversity metric in selecting appropriate members of neural network ensembles. For comparison purpose, we also evaluate an existing method named GASEN.

In regression analysis, multicollinearity between independent variables can affect the variance of the estimated regression coefficients severely. Pair-wise correlations between predictors, t-tests and F-test, are some of the common methods used for detecting multicollinearity. Another method which is favored by

many regression analysts is the use of variance inflation factors (VIF).

Variance inflation factor is a statistical measure that quantifies the severity of multicollinearity of the i^{th} independent variable with the other independent variables, in regression analysis. More specifically, it quantifies how much the variances of the estimated regression coefficients are inflated. In [22], Greene derived the variance-covariance matrix of the regression coefficients as:

$$\sigma(b_i b_j) = \sigma_\epsilon^2 (\mathbf{X}^T \mathbf{X})^{-1} \quad (1)$$

where \mathbf{X} is an n by $k+1$ matrix with the first column consisting of ones and the next k columns consisting of the σ_ϵ^2 values of k independent variables, \mathbf{X}^T is the transpose of \mathbf{X} , and σ_ϵ^2 is the population variance of the residuals. Based on Eq. (1), Robert [23] derived an equation that provides the unbiased estimate of the variance of the i^{th} regression coefficient as:

$$\hat{\sigma}^2(b_i) = \frac{\left[\frac{(1 - R_y^2) \times \sum (\mathbf{Y}_i - \bar{\mathbf{Y}})^2}{(n - k - 1)} \right]}{(1 - R_i^2) \times \sum \mathbf{x}_i^2} \quad (2)$$

where n is the sample size, k the number of independent variables in the regression analysis, \mathbf{x}_i are the independent variables (the last $k-1$ columns of \mathbf{X} in Eq. (1)), \mathbf{Y}_i is the dependent variable (generally one dependent variable is used), R_i^2 is the proportion of the variance in the i^{th} independent variable that is associated with the other independent variables in the analysis. R_y^2 is the squared multiple correlation of the dependent variable regressed on all other independent variables in the analysis.

Rearranging Eq. (2), we get,

$$\hat{\sigma}^2(b_i) = \frac{\left[\frac{(1 - R_y^2) \times \sum (\mathbf{Y}_i - \bar{\mathbf{Y}})^2}{(n - k - 1)} \right]}{\sum \mathbf{x}_i^2} \times \frac{1}{(1 - R_i^2)} \quad (3)$$

The second term on the right hand side of Eq. (3) represents the VIF of the i^{th} independent variable, and it indicates the multiplicative increase in the variance of the regression coefficient of this variable [23]. It is worth mentioning that the variance inflation factor VIF_i of the i^{th} independent

variable can be found by regressing it on the other independent variables.

In our case, the independent variables are the validation errors of the ensemble members (i.e. the base networks). Thus, for the n^{th} base network VIF_n is defined as follows.

$$VIF_n = \frac{1}{(1 - R_n^2)} \quad (4)$$

Where VIF_n is the variance inflation factor for the n^{th} network, and R_n^2 is the R^2 value obtained by regressing the n^{th} independent variable (the validation errors of the n^{th} network) on the remaining independent variables (i.e. the validation errors of the others networks).

There are many rules of thumb associated with VIF that are regarded as a sign of severe multicollinearity. The most commonly used is the rule of 10, that is if $VIF_n > 10$, the n^{th} network has serious multicollinearity with the other ensemble members, otherwise there is less or even no multicollinearity [23]. In this paper, instead of considering any threshold value, we try to minimize the sum of all VIF_n . The value of this sum is considered as our second objective in section 2.3. For detailed discussion on VIF we refer interested readers to [23, 24].

2.2.2 Training Component Networks

For each data set, bootstrap sampling was used on the original training data to generate 50 new training data. Each of these new training data is then used to train a component network (or base network), using back-propagation algorithm with different initial weights and with early-stopping option (on the validation data). Early stopping is a method to improve the generalization capability of ANN in case of small size training data; the training is stopped when the error on the validation set has reached a certain threshold. Fifty single-hidden-layered base networks with similar structure (8:5:1, that is 8 input neurons, 5 hidden neurons, and one output neuron) were trained. These networks constitute the pool of base networks on which we applied two selection based methods: GASEN, and MOGASEN. For the standard bagging method, all the base networks are considered as component networks of the ensemble, and the output of the ensemble is obtained by averaging the outputs from these base networks; whereas only outputs from the selected networks are considered in other methods.

2.2.3 Best Ensemble Selection

Two selection-based methods are used to select the best ensemble: GASEN, and MOGASEN. For GASEN method, default settings specified by the authors [25] were used, except for the number of generations and the population size. For MOGASEN method, MATLAB implementation of MOGA was used. Table 1 shows settings of some important parameters in MOGA; default settings were used for other parameters.

Table 1: MOGA parameters setting

Parameter	Value/Scheme
Population Size	50
Population type	Bit String
Number of variables	50
Generations	50
Mutation probability	0.2
Crossover probability	0.8
Selection	Tournament

MOGASEN based selection process is performed through four steps as described below.

Individual encoding: To solve our optimization problem, a solution is first encoded to chromosome form, the size of the search space is the same as the number of primary base networks, K . Binary encoding scheme is used, wherein 0 means the base network is excluded and 1 means the base network is selected. For instance, if chromosome $C = 10101011$ (when $K = 8$) means that the base learners #1, #3, #5, #7, and #8 are selected as members of the ensemble ($h = 5$).

Initial population: The initial population is randomly generated.

Objective functions: Two objective functions were used, the ensemble error and the diversity metric (VIF). More specifically, let's call these objective functions f_1 , and f_2 , respectively. Then we can define them as follows.

$$f_1 = 100 \times \frac{1}{N} \sum_{i=1}^N \frac{|(y_i - y_i^{ens})|}{y_i} \quad (5)$$

where N is the number of validation samples, y_i is the actual value of the i^{th} sample and y_i^{ens} is its predicted value by the ensemble. It is worth mentioning that y_i^{ens} is the average value of the outputs from all the selected base

networks. For instance, if we have K base networks y_i^{ens} is obtained using Eq. (6).

$$y_i^{ens} = \frac{1}{K} \sum_{k=1}^K y_i^k \quad (6)$$

where y_i^k is the output from the k^{th} network for the i^{th} sample.

$$f_2 = \sum_{n=1}^K VIF_n \quad (7)$$

where K is the number of base networks in a potential solution, and VIF_n is the VIF of the n^{th} base network, as defined in Eq. (4).

Genetic operations: In standard GA, three operations are generally involved: selection, crossover, and mutation. During the selection step, chromosomes with highest fitness values are chosen as parents. From these parents, candidates (children, offspring) are generated using crossover and mutation operations. The algorithm calculates a fitness score for each candidate and replaces chromosomes with low scores by new candidates with high scores. This process is repeated until stopping conditions (maximum number of generations, a certain value of the fitness function, etc...) are satisfied.

The proposed approach is summarized in Table 2, where D_{Tr} is the original training data; D_V is the validation data used during MOGA based base networks selection, D_{Te} is the test data which is only used after ensemble construction, and B is the number of bootstrap replicates which is also equal to the number of initial base networks (in our case 50 base networks).

Table 2: MOGASEN method

<p>Base networks generation:</p> <ol style="list-style-type: none"> 1) Use bootstrap sampling on D_{Tr} to generate B new training sets 2) Use each of the new training sets obtained in 1) to train a base network using BP with early-stopping (against the validation data) <p>Best ensemble selection:</p> <ol style="list-style-type: none"> 3) Randomly generate an initial population 4) Use MOGA with two objective functions as defined in Eqs. (5) & (7) to evolve the initial population. At termination, trace out the solution with the smallest f_2 on D_V and report it as the selected ensemble. 5) Evaluate the ensemble selected in 4) on D_{Te}, compute the test error using Eq. (5) with N as the number of test samples instead.

3. Results and Discussion

All computations were carried out using MATLAB R2010a (MathWorks Inc.) software on a desktop computer with Intel(R) Core(TM) i3 T2450 2.93 GHz CPU, 2 GB RAM and Windows XP professional operating system. It is worth mentioning the following notations: BNN for the best base network, Bagging for the standard bagging method, GASEN for GASEN method, and MOGASEN for our method.

To avoid biased comparison, for each method and each data set we perform ten runs and recorded the averages of mean absolute percentage errors(MAPE) for the selected ensembles as well as for the best base networks. Experimental results are reported in Tables 3 and 4. Also, for comparison purpose, results of standard bagging are shown.

When constructing ensemble of predictors, the capability of a method in reducing the size of an ensemble while maintaining or even improving its performance is also of great importance. Having this in mind, we computed the average number of selected base networks over ten runs, for GASEN and MOGASEN. Results are reported in Table 5.

Table 3: Averaged validation errors over ten runs

Data sets	Methods			
	BNN	Bagging	GASEN	MOGASEN
Formaldehyde	(31.2425)(40.0723)(35.9525) ^a	41.3719	40.9267	34.1758
Benzene	(8.3587)(8.4771)(8.5293)	9.1427	7.0157	7.6037
Toluene	(12.2525)(12.2525)(12.96045)	19.5845	11.6776	13.74269
Carbon monoxide	(15.3124)(16.5098)(17.67203)	42.8593	18.0637	16.52416

^aNumbers in parentheses are errors of best component networks for bagging, GASEN, MOGASEN, in order

Table 4: Averaged test errors over ten runs

Data sets	Methods			
	BNN	Bagging	GASEN	MOGASEN
Formaldehyde	(36.5643)(74.9503)(54.7101) ^a	63.5795	57.5075	40.1511
Benzene	(7.9458)(14.0323)(13.6309)	9.0549	11.7958	10.48934
Toluene	(13.542)(16.5326)(16.26096)	17.9958	11.9847	14.09098
Carbon monoxide	(14.8845)(18.1362)(19.0778)	36.5611	23.0379	18.60605

^aNumbers in parentheses are errors of best component networks for bagging, GASEN, MOGASEN, in order

Table 5: Average number of selected nets over ten runs

Data sets	Methods	
	GASEN	MOGASEN
Formaldehyde	5	5
Benzene	7	5
Toluene	5	5
Carbon monoxide	5	5

From Tables 3 and 4 one can notice that the theory of “many could be better than all” was verified. In most cases, GASEN and MOGASEN outperformed the standard bagging and their corresponding best base networks, with unique exception on benzene. The only case where GASEN performs better than MOGASEN on both validation and test data sets is with toluene. An intuitive remark from these results is that an ensemble less effective than the best component network on a given data set (here, validation data set) may perform well on a novel data set (e.g. GASEN on formaldehyde data set). Also, results from carbon monoxide data set infer that there were too many redundant networks in the initially generated pool of base networks. This resulted in poor performance of standard bagging method which is normally known to be effective on component networks that are sufficiently accurate and diverse. Indeed, being the smallest data set with almost 60 samples, using simple bootstrapping and different initial weights on such small data set to train base networks was insignificant for generating diverse networks. This is in

perfect agreement with results obtained in our recent work [26], where base networks with different topologies were even used. GASEN tends more to selecting best networks than diverse ones. This can be seen with benzene and toluene data sets. A possible reason for this is the value of the threshold λ . Setting the threshold to high values will cause the algorithm to only emphasize on accurate base networks, while setting it to small values will result in selection of inaccurate networks.

Results from Table 5 show that GASEN and MOGASEN selected almost the same number of component networks over ten runs, except with benzene data set where the average number of selected networks by GASEN is superior to that selected by MOGASEN. By selecting more component networks in the case of benzene, GASEN might have overfitted the validation data, thereby resulting in lower performance on test data compared to MOGASEN. Another important remark is that, as both GASEN and MOGASEN are off-line selection methods, one may suspect that component networks selected by these methods are in fact similar. However, results from Tables 3 and 4 evidenced that this rarely happened in practice.

4. Conclusion

Diversity metrics (pair-wise as well as non-pairwise) play an important role in ensemble learning. In this paper, a new non-pairwise diversity metric based on variance inflation factor is proposed. A multi-objective GA with two objectives (ensemble error and the new diversity metric) is

used to select best neural network ensembles for concentration estimation of some indoor air pollutants. Empirical results show that variance inflation factor can effectively be used as diversity metric. Although the proposed method can outperform GASEN (another selection based method), standard bagging, and the best component network; more study on VIF for multicollinearity measurement is required to further improve the method.

This method is not restricted to electronic nose data; it can be applied in other fields. Also, it can be extended to classification problem. In this paper small size data sets were used, we therefore need to evaluate this method on large-scale data sets (for both classification and regression); this will be considered in our future work.

Acknowledgments

The authors would like to acknowledge financial supports from the Key Science and Technology Research Program (CSTC2010AB2002, CSTC2009BA2021), the Chongqing University Postgraduate Science and Innovation Fund (CDJXS12160005), and the Fundamental Research Funds for the Central Universities of China (CDJXS10161114). Special thanks to the anonymous reviewers for their valuable comments and suggestions.

References

[1] L.K. Hansen, P. Salamon, "Neural network ensembles", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 12, No.10,1990,pp.993-1001.
[2] Z.-H. Zhou, Y. Jiang, Y.-B. Yang, S.-F. Chen, "Lung cancer cell identification based on artificial neural network ensembles", Artificial Intelligence in Medicine, Vol. 24, No.1, 2002, pp. 25-36.
[3] G. Daqi, C. Wei, "Simultaneous estimation of odor classes and concentrations using an electronic nose with function approximation model ensembles", Sens. Actuators, B Chem. Vol.120, 2007, pp. 584-594.
[4] Evandro Bona, et al., "Optimized Neural Network for Instant Coffee Classification through an Electronic Nose", International Journal of Food Engineering: Vol. 7, Issue 6, 2011.
[5] L.K. Hansen, L. Liisberg, P. Salamon, "Ensemble methods for handwritten digit recognition", in Proceedings IEEE Workshop on Neural Networks for Signal Processing, 1992, pp.333-342.
[6] L. Breiman, "Bagging predictors", Machine Learning, Vol. 24, No. 2, 1996, pp.123-140.
[7] Opitz, D. and Maclin, R., "Popular ensemble methods: An empirical study", Journal of Artificial Intelligence Research, Vol. 11, 1999, pp.169-198.
[8] A. Krogh, J. Vedelsby, "Neural network ensembles, cross validation, and active learning", Advances in Neural Information Processing Systems, Vol. 7, 1995, pp.231-238.

[9] D.W. Opitz, J.W. Shavlik, "Generating accurate and diverse members of a neural network ensemble", Advances in Neural Information Processing Systems, Vol. 8, 1996, pp.535-541.
[10] X. Yao, Y. Liu, "Making use of population information in evolutionary artificial neural networks", IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics Vol.28, No. 3, 1998, pp.417-425.
[11] Zhi-Hua Zhou et al., "Ensembling Neural Networks: Many Could Be Better Than All", Artificial Intelligence, Vol.137, No.1-2, 2002, pp. 239-263.
[12] R. J. Mammone, Artificial Neural Networks for Speech and Vision, London: Chapman & Hall, pp.126-142, 1993.
[13] D.W. Opitz, J.W. Shavlik, "Actively searching for an effective neural network ensemble", Connection Science Vol.8, No.3-4, 1996, pp. 337-353.
[14] C.J. Merz, M.J. Pazzani, "Combining neural network regression estimates with regularized linear weights", Advances in Neural Information Processing Systems, Vol. 9, 1997, pp.564-570.
[15] D. Jimenez, "Dynamically weighted ensemble neural networks for classification", in Proceedings IJCNN-98, vol. 1, 1998, pp.753-756.
[16] N. Ueda, "Optimal linear combination of neural networks for improving classification performance", IEEE Transaction Pattern Analysis and Machine Intelligence, Vol. 22, No. 2, 2000,pp. 207-215.
[17] L. Zhang et al., "Classification of multiple indoor air contaminants by an electronic nose and a hybrid support vector machine", Sensors and Actuators B, Vol.174, 2012, pp.114-125.
[18] L.Zhang et al., "Chaos based neural network optimization for Concentration Estimation of Indoor Air Contaminants", Sens. and Actuat. A, Vol.189, 2013, pp.161-167.
[19] A. Szczurek et al., "The stop-flow mode of operation applied to a single chemiresistor", Sensors and Actuators B, Vol. 148, 2010, pp.522-530.
[20] R.W. Kennard, L.A. Stone, "Computer aided design of experiments", Technometrics, Vol. 11, 1969, pp. 137-148.
[21] Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison- Wesley Pub. Co., 1989.
[22] Greene, W. H., Econometric Analysis, 2nd ed. Macmillan, New York, 1993.
[23] Robert, M. O'Brien, "A Caution Regarding Rules of Thumb for Variance Inflation Factors". Qual. Quant. 41, 2007, pp.673-690.
[24] Kutner, Nachtsheim, Neter, Applied Linear Regression Models, fourth ed. McGraw-Hill, Irwin, 2004.
[25] Zhi-Hua Zhou, et al., "Genetic Algorithm based Selective Neural Network Ensemble", in Proceedings of the 17th International Joint Conference on Artificial Intelligence, 2001, vol.2, pp.797-802.
[26] Chaibou Kadri, Fengchun Tian, Lei Zhang, Guojun Li, Lijun Dang and Guorui Li, "Neural Network Ensembles for Online Gas Concentration Estimation Using an Electronic Nose", International Journal of Computer Science Issues, Vol. 10, Issue 2, No. 1, 2013, pp.129-135.