Convergence Analysis of Backpropagation Algorithm for Designing an Intelligent System for Sensing Manhole Gases

Article · August 2016

DOI: 10.1007/978-81-322-2544-7_7

CITATION
1

READS
60

4 authors:

Varun Kumar Ojha
ETH Zurich
40 PUBLICATIONS **91** CITATIONS
SEE PROFILE

Paramartha Dutta
Visva Bharati University
158 PUBLICATIONS **666** CITATIONS
SEE PROFILE

Atal Chaudhuri
Jadavpur University
45 PUBLICATIONS **101** CITATIONS
SEE PROFILE

Hiranmay Saha
Indian Institute of Engineering Science and T…
257 PUBLICATIONS **2,354** CITATIONS
SEE PROFILE

Convergence Analysis of Backpropagation Algorithm for Designing an Intelligent System for Sensing Manhole Gases

Varun Kumar Ojha, Paramartha Dutta, Atal Chaudhuri and Hiranmay Saha

Abstract Human fatalities are reported due to excessive proportional presence of hazardous gas components in manhole, such as hydrogen sulphide, ammonia, methane, carbon dioxide, nitrogen oxide, carbon monoxide, etc. Hence, predetermination of these gases is imperative. A neural network (NN)-based intelligent sensory system was proposed for avoiding such fatalities. Backpropagation (BP) was applied for supervised training of the proposed neural network model. A gas sensor array consists of many sensor elements was employed for sensing manhole gases. Sensors in the sensor array were only responsible for sensing their target gas components. Therefore, the presence of multiple gases results in cross sensitivity that is a crucial issue to this problem. It is viewed as a pattern recognition and noise reduction problem. Various performance parameters and complexity of the problem influences NN training. In this chapter, performance of BP algorithm on such real-life application was comprehensively studied, compared, and contrasted with several hybrid intelligent approache, both in theoretical and in statistical senses.

Keywords Gas detection · Backpropagation · Neural network · Pattern recognition · Parameter tuning · Complexity analysis

V.K. Ojha (✉) · A. Chaudhuri
Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
e-mail: varun.kumar.ojha@vsb.cz

A. Chaudhuri
e-mail: atalc23@gmail.com

V.K. Ojha
IT4Innovations, VŠB Technical University of Ostrava, Ostrava, Czech Republic

P. Dutta
Department of Computer and System Sciences, Visva-Bharati University, Santiniketan, India
e-mail: paramartha.dutta@gmail.com

H. Saha
Centre of Excellence for Green Energy and Sensors System, Indian Institute of Engineering Science and Technology, Howrah, India
e-mail: sahahiran@gmail.com

S. Bhattacharyya et al. (eds.), *Hybrid Soft Computing Approaches*,
Studies in Computational Intelligence 611,
DOI 10.1007/978-81-322-2544-7_7

1 Introduction

Computational Intelligence (CI) offers solutions to a wide range of real-life problems. In this chapter, we have resorted to using a CI approach to offer a design of an intelligent sensory system (ISS) for detecting manhole gas mixture. The manhole gas mixture detection problem was treated as a pattern recognition/noise reduction problem. For the past decades, neural network (NN) has been proven as a powerful tool for machine learning application in various domains. In this chapter, we have used backpropagation (BP) NN technique for designing an ISS. The primary aim of this chapter is to provide a comprehensive performance study of BP algorithm.

Decomposition of wastage and sewage into sewer pipeline leads to toxic gaseous mixture formation, known as a manhole gas mixture that usually has toxic gases such as hydrogen sulphide (H_2S), ammonia (NH_3), methane (CH_4), carbon dioxide (CO_2), nitrogen oxides (NO_x), etc., [1–3]. Often, human fatalities occur due to excessive proportion of the mentioned toxic gases. The persons, who have the responsibilities for the maintenance and cleaning of sewer pipeline are in need of a compact instrument that may able predetermine safeness of manholes. In the recent years, due to the toxic gas exposures, several instances of deaths were reported, including municipality labourers and pedestrians [4–8]. We have investigated the commercially available gas sensor tools. We found that the commercially available gas detectors are insufficient in sensing all the aforementioned gases and the cross sensitive in their response is a basic problem associated with these sensor units.

Rest of the chapter is organized as follows: Sect. 2.1 provides a brief literature survey followed by a concise report on our research contribution in Sect. 2.2. Readers may find discussion on design issues of the proposed intelligent sensory system (ISS) in Sect. 2.3, a discussion on data sample preparation in Sects. 2.5 and the crucial issue of the cross sensitivity in Sect. 2.4. Section 2.6 provides a brief discussion on NN configuration, training pattern and BP algorithm. Performance analysis of BP algorithm on such real application provided in Sect. 3 is central subject of this chapter. Finally, Sections 4 and 5 provide results and conclusion, respectively.

2 Mechanism

This section provides a detailed discussion on the various materials and methods acquired in the design and the development of the ISS. We thoroughly explains design issues of an intelligent gas detection system, data collection process and data preparation technique.

2.1 A Brief Literature Survey

In the past few years, several research work have been reported on the development and design of electronic nose (E-NOSE) and gas detection system. After an detailed literature examination, we may appreciate the effort by Li et al. [9] for his contribution in developing a NN-based mixed gas (NO_x, and CO) measurement system. On the other hand, Sirvastava et al. [10, 11] have proposed a design of intelligent E-NOSE system using BP and Neuro-genetic approach. A pattern recognition technique, based on the wallet transformation for gas mixture analysis using single tin-oxide sensor was presented by Llobet [12]. Liu et al. [13] addressed a genetic-NN algorithm to recognize patterns of mixed gases (a mixture of three component gases) using infrared gas sensor. Tsirigotis et al. [14] illustrated a NN-based recognition system for CO and NH_3 gases using metallic oxide gas sensor array (GSA). Lee et al. [15] illustrated uses of micro-GSA combined with NN for recognizing combustible leakage gases. Ambard et al. [16] have demonstrated use of NN for gas discrimination using a tin-oxide GSA for the gases H_2, CO and CH_4. In [17], authors have illustrated a NN-based technique for developing a gas sensory system for sensing gases in a dynamic environment. Pan et al. [18] have shown several applications of E-NOSE. Wongchoosuka et al. [19] have proposed an E-NOSE detection system based on carbon nanotube-SnO_2 gas sensors for detecting methanol. Zhang et al. [20] developed a knowledge-based genetic algorithm for detecting mixed gas in mines. Won and Koo [21] proposed a system for estimation of hazardous gas release rate using optical sensor and NN-based technique. A comprehensive study of the above-mentioned articles lead us to the following conclusion: (i) mostly, BP and NN-based techniques were used for gas detection problem in the respective application areas; (ii) mainly, two or three gas mixture detection was addressed and that too, the gases were those, whose sensors were not cross sensitive at high extent; and (iii) cross sensitivity was not addressed firmly. In the design of manhole gas mixture detection system, cross sensitivity due to the presence of several toxic gases is a vital issue. Present article firmly addresses this issue. Ojha et al. [22–24] have presented several approaches towards potential solution to manhole gas detection issue.

2.2 Present Approach and Contribution

In this chapter, manhole gas detection problem was treated as a pattern recognition/noise reduction problem, where a NN regressor was modelled and trained in supervised mode using BP algorithm. A semiconductor-based GSA containing distinct semiconductor-type gas sensors was used to sense the presence of gases according to their concentration in manhole gas mixture. Sensed values were

cross sensitive as multiple gases exists in manhole gas mixture. Cross sensitivity was occurred because the gas sensors are sensitive towards non-target gases too. Our aim was to train a NN regressor such that cross sensitivity effect can be minimized. The developed ISS would help persons to be watchful against the presence of toxic gases before entering into manholes and thereby avoiding fatality. Various parameters of BP algorithm were tuned to extract-out best possible result. Performance of BP on various parameters tuning are comprehensively reported. In addition, performance analysis of BP algorithm against various hybrid intelligent approaches such as conjugate gradient, neuro-genetic (NN trained using genetic algorithm), and neuro-swarm (NN trained using particle swarm optimization algorithm) is reported, both in theoretical and statistical senses.

2.3 Basic Design of Intelligent Sensory System

The design illustrated in this chapter consisted of three constituent units: input, intelligent and output. Input unit consists of gas suction-motor chamber, GSA and data acquisition-cum data-preprocessor block. Intelligent unit receives data from input unit and after performing its computation, it sends result to output unit. Output unit presents systems' output in user-friendly form. Gas mixture sample is allowed to pass over semiconductor-based GSA. Preprocessing block receives sensed data values from GSA and make sure that the received data values are normalized before feeding it to NN. Later, output unit perform de-normalization of network response to generates alarm, if any of the toxic gas components exceeds their safety limit. For training of NN, several data samples were prepared. Block diagram shown in Fig. 1 is a lucid representation of above discussion.

2.4 Semiconductor-Based Gas Sensor Array (GSA) and Cross Sensitivity Issue

Metal oxide semiconductor (MOS) gas sensors were used for GSA. We used N distinct sensor element representing n gases. MOS sensors are resistance-type electrical sensors, where response was change in circuit resistance proportional to gas concentration, given as $\Delta R_s/R_0$, where ΔR_s is change in MOS sensor resistance and R_0 is base resistance value [15, 19]. A typical arrangement of GSA is shown in Fig. 2. The circuitry shown in Fig. 2 was developed in our laboratory. In Ghosh et al. [25, 26], we elaborately discussed sensor array and its working principles. Although gas sensor elements were supposed to detect only their target gases, they showed sensitivity towards other gases too. Hence, sensor array response always involved cross-sensitivity effect [9]. This indicates that sensors' responses were noisy. If we concentrate on the first and second rows in

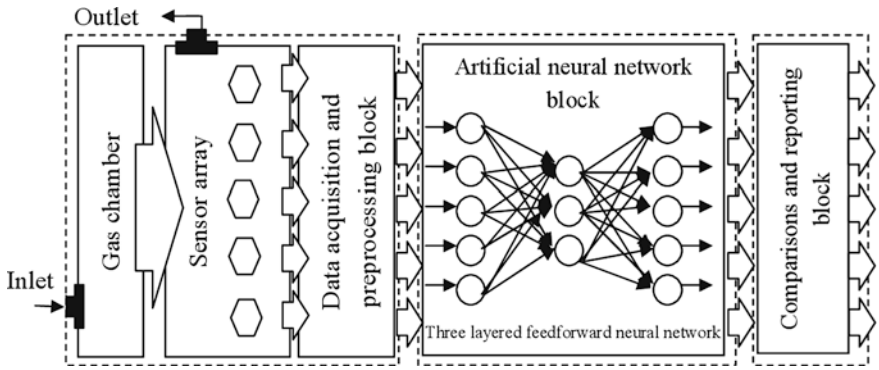

Fig. 1 Intelligent sensory system for manhole gas detection

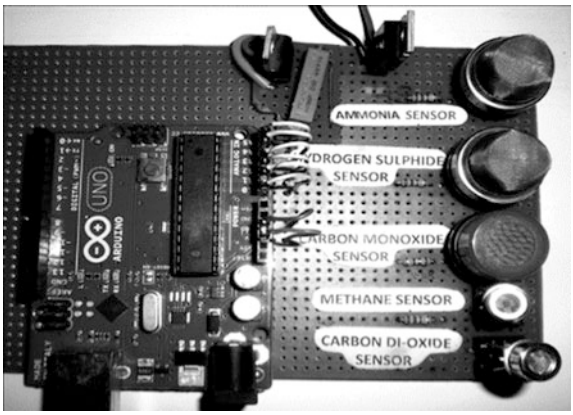

Fig. 2 Gas sensor array with data acquisition system for sensing concentration of manhole gases

Table 1, we may observe the inherent cross-sensitivity effect in responses of sensors. The first and second samples in Table 1 indicate that changes in concentration of only methane gas resulted in change in response of all the other sensors, including the sensor earmarked for methane gas. It is indicated that the prepared data sample was containing cross-sensitive effect. We may observed that cross-sensitivity effect was not random, rather followed some characteristics and patterns. Hence, in operating (real-world) environment, raw sensor responses of GSA may not be used for predicting concentration of manhole gases. Therefore, to predict/forecast level of concentration of manhole gases, we proposed to use ISS, equipped with pattern recognition/noise reduction techniques that will help to filter-out noise induced on sensors due to cross sensitivity.

Table 1 Data sample for intelligent sensory system (samples illustrated for example purpose)

No. of Sample	Sample gas mixture (in ppm)					Sensor response ($\Delta R_s/R_0$)				
	NH_3	CO	H_2S	CO_2	CH_4	NH_3	CO	H_2S	CO_2	CH_4
1	50	100	100	100	**2000**	0.053	0.096	0.065	0.037	0.121
2	50	100	100	100	**5000**	**0.081**	**0.108**	**0.074**	**0.044**	**0.263**
3	50	100	100	200	2000	0.096	0.119	0.092	0.067	0.125
4	50	100	200	200	5000	0.121	0.130	0.129	0.079	0.274
5	50	100	200	400	2000	0.145	0.153	0.139	0.086	0.123

2.5 *Data Collection Method*

Data sample for experiment and NN training was prepared in following steps: In the first step, we collected information about safety limits of the component gases of manhole gas mixture. After that, distinct concentration values (level) around the safety limits of each manhole gas were recognized. Several gas mixture samples were prepared by mixing gas components in different combinations of their concentration level. As an example, if we have five gases and three recognized concentration levels of each gas, then, we may mix them in 243 different combinations. Hence, we obtained 243 gas mixture samples. When these samples were allowed one by one to pass over semiconductor-based GSA, as a result, we got data table shown in Table 1.

2.6 *Neural Network Approach*

We already mentioned in Sect. 2 that raw sensor response may not accurately represent real-world scenario. Therefore, we were inclined to use NN technique in order to reduce noise, so that prediction can achieve high accuracy with lowest possible error.

2.6.1 Multi-layer Perceptron

NN, "a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for subsequent use" [27], trained using BP algorithm may offer a solution to aforementioned problem. NN shown in Fig. 3 contains 5 input nodes, n hidden nodes with l layers and 5 output nodes leading to a $5 - n \ldots n - 5$ network configuration. Nodes in input layer as well as in output layer indicate that our system was developed for detecting 5 gases from manhole gaseous mixture. A detailed discussion on NN performance based on network configuration is provided in Sect. 3.1.2.

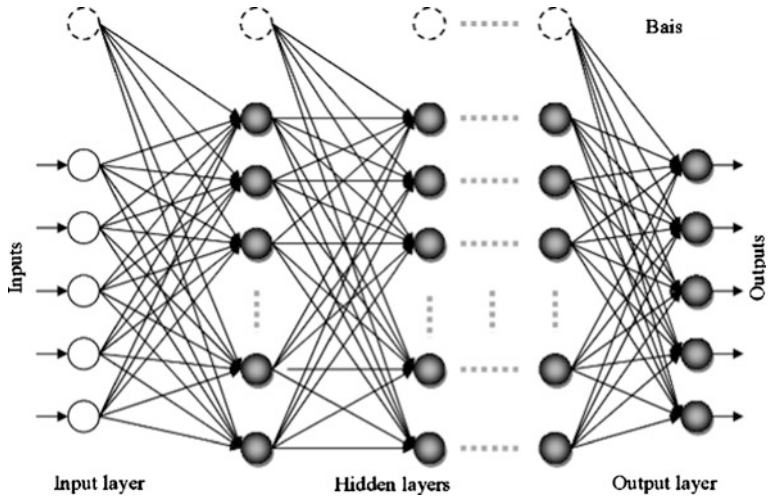

Fig. 3 NN architecture for five input manhole gas components

2.6.2 Training Pattern

We acquired supervised mode of learning for training of NN. Hence, the training pattern constituted of input vector and target vector. We mentioned above that normalized sensor responses were given as input to NN. Hence, input vector I consisted of normalized values of sensor responses. In given data sample, input vector was a five-element vector, where each element represents a gas of gas mixture. Input vector I can be represented as follows:

$$I = [i_1,\ i_2,\ i_3,\ i_4,\ i_5]^T. \tag{1}$$

The system output was presented in terms of concentration of gases. Hence, target vector T was prepared using values of gas mixture sample. In given data sample, target vector was a five-element vector, where each element represents a gas in gas mixture. Target vector T can be represented as follows:

$$T = [t_1, t_2, t_3, t_4, t_5]^T. \tag{2}$$

A training set containing input vector and target vector can be represented as per Table 2.

Table 2 Training set for neural network

No. of pattern	Input vector I					Target vector T				
	i_1	i_2	i_3	i_4	i_5	t_1	t_2	t_3	t_4	t_5
(I_1, T_1)	0.19	0.35	0.23	0.13	0.44	0.01	0.02	0.02	0.02	0.4
(I_2, T_2)	0.29	0.39	0.27	0.16	0.96	0.01	0.02	0.02	0.02	1
(I_3, T_3)	0.35	0.43	0.33	0.24	0.45	0.01	0.02	0.02	0.04	0.4
(I_4, T_4)	0.44	0.47	0.47	0.28	1	0.01	0.02	0.04	0.04	1
(I_5, T_5)	0.52	0.55	0.51	0.31	0.45	0.01	0.02	0.04	0.08	0.4

2.6.3 Backpropagation (BP) Algorithm

Let us have a glimpse of BP algorithm as described by Rumelhart [28]. BP algorithm is a form of supervised learning for training of multi-layer NNs, also known as generalized delta rule [28], where error data from output layer are backpropagated to earlier ones, allowing incoming weights to be updated [27, 28]. Synaptic weight matrix W can be updated using delta rule is as:

$$W(n+1) = W(n) + \Delta W(n+1), \tag{3}$$

where n indicates nth epoch training and $\Delta W(n+1)$ is computed as:

$$\Delta W(n+1) = \eta g(n+1) + m \cdot g(n), \tag{4}$$

where η is learning rate, β is momentum factor, and $g(n)$ is gradient of nth epoch computed as:

$$g(n) = \delta_j(n) y_i(n). \tag{5}$$

Local gradient $\delta_j(n)$ is computed both at output layer and hidden layer as follows:

$$\begin{aligned} \delta_j(n) &= -e_j(n)\varphi\big(v_j(n)\big) \text{ for output layer} \\ &= \varphi\big(v_j(n)\Sigma_k \delta_k(n) w_{kj}(n)\big) \text{ for hidden layer.} \end{aligned} \tag{6}$$

The algorithm terminates either sum of squared error (SSE) reached to an acceptable minimum, or algorithm completes its maximum allowed iterations. SSE measures performance of BP algorithm [29], computed as:

$$\text{SSE} = \frac{1}{2}\sum\nolimits_{\text{p}} \big(O_{\text{pi}} - t_{\text{pi}}\big)^2 \text{ for all } p \text{ and } i, \tag{7}$$

where O_{pi} and t_{pi} are actual and desired outputs, respectively, realized at output layer, p is total number of input pattern vectors and i is number of nodes in output layer. A flow diagram shown in Fig. 4 clearly illustrates aforementioned BP algorithm.

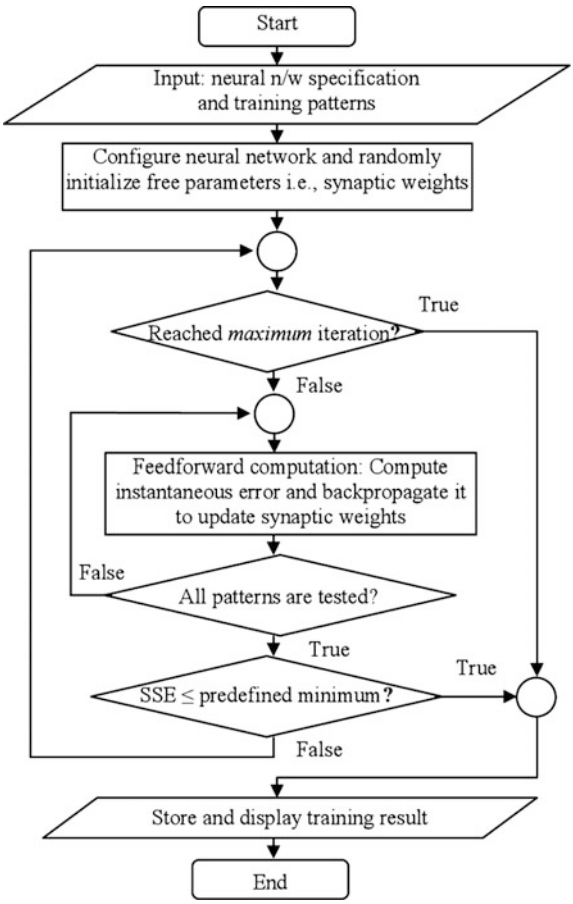

Fig. 4 Schematic of BP algorithm flow diagram

3 Performance Study Based on Various Parameters

We implement BP algorithm using JAVA programming language. Training of NN model was provided by using data sample prepared as per method indicted in Table 2. Thereafter, performance of BP algorithm was observed. An algorithm used for training of NN for any particular application is said to be efficient if and only if, SSE or mean square error (MSE) induced on NN for given training set can be reduced to an acceptable minimum. BP algorithm is robust and popular algorithm used for training of multi-layer perceptrons (MLPs). Performance analysis presented in this section aims to offer an insight of the strengths and weaknesses of BP algorithm used for the mentioned application problem. Performance of BP algorithm depends on adequate choice of various parameters used in algorithm and complexity of problem for which the algorithm used. We may not control the

complexity of the problem, but we may regulate the underlying parameters in order to enhance BP algorithm's performance. Even though BP algorithm is widely used for NN training, its several controlling parameters are one of the reasons that motivated research community to think of the alternatives of BP algorithm. Our study illustrates influence of these parameters over the performance of BP algorithm.

3.1 Parameter Tuning of BP Algorithm

Performance of BP algorithm is highly influenced by various heuristics and parameters used [27, 28]. These includes training mode, network configuration (network model), learning rate (η), momentum factor (β), initialization of free parameters (synaptic weights), i.e. initial search space (χ), volume of training set, and terminating criteria.

3.1.1 Mode of Training

BP algorithm has two distinguish modes of training: sequential mode and batch mode. Both of these training methods have their advantages and disadvantages. In sequential mode, free parameters (synaptic weights) of NN are made to be adjusted for each of training example of a training set. In other words, synaptic weight adjustment is based on instantaneous error induced on NN for each instance of training example. This particular fashion of weight adjustment makes the sequential mode training easy and simple to implement. Since the sequential mode training is stochastic in nature, convergence may not follow smooth trajectory, but it may avoid being stuck into local minima and may lead to a global optimum if one exists. On contrary, in batch mode, free parameters are updated once in an epoch. An epoch of training indicates the adjustment of free parameters of a NN that takes place once for entire training set. In other words, training of NN is based on SSE induced on NN. Hence, gradient computation in this particular fashion of training is more accurate. Therefore, the convergence may be slow, but may follow smooth trajectory. Figures 5 and 6 indicate performance of BP algorithm based on these two distinct modes of NN training. Figure 5 indicates that the convergence using batch mode training method was slower than that of sequential mode training. From Figs. 5 and 6, we may be observed that sequential mode training method may not follow smooth trajectory, but it may lead to global optimum, whereas, it is evident that in batch mode training method, convergence was following smooth trajectory.

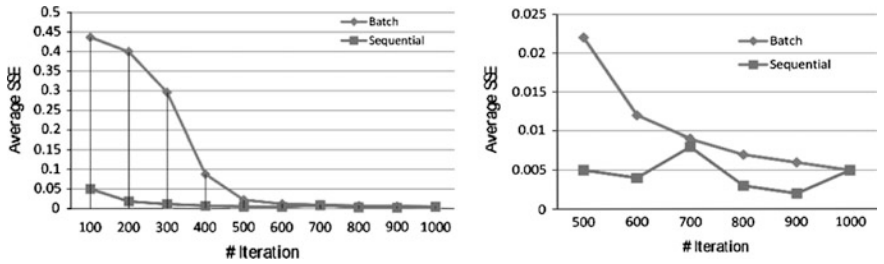

Fig. 5 Convergence: sequence versus batch mode

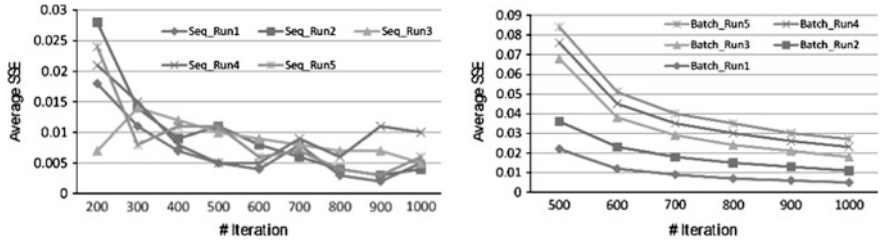

Fig. 6 Convergence trajectory in (*left*) sequence mode and (*right*) batch mode

3.1.2 Network Configuration

Multi-layer perceptrons are used for solving nonlinear problems. Basic MLP configuration consists of one input layer, one output layer with one or more hidden layer (s). Each layer may consist of one or more processing unit (neurons). The primary task in devising problems in NN is the selection of appropriate network model. Finding an optimum NN configuration is also known as structural training of NN. Network selection process starts with the selection of most basic three-layer architecture. Three-layer NN consists of a input layer, a hidden layer and a output layer. Number of neurons at input and output layers depends on problem itself. The gas detection problem was essentially a noise reduction problem, where NN tries to reduce noise of signals emitting from sensors in the presence of gas mixture. Hence, it was obvious that the number of outputs equals the number of inputs. In this application, we were designing a system that detects five gases. It is because there were five input signals to NN that leads to five processing units at input layer and five processing units at output layer. In three-layer NN configuration, the network configuration optimization reduces to the scope of regulating number of processing units (neurons) at hidden layer. Keeping other parameters fixed to certain values, the number of nodes at hidden layer was regulated from 1 to 8 to monitor influence of NN configuration on performance of BP algorithm. Parameter setup was as follows: number of iterations was set to thousand, η to 0.5, β to 0.1, $\chi \in [-1.5, 1.5]$ and training volume to fifty training examples.

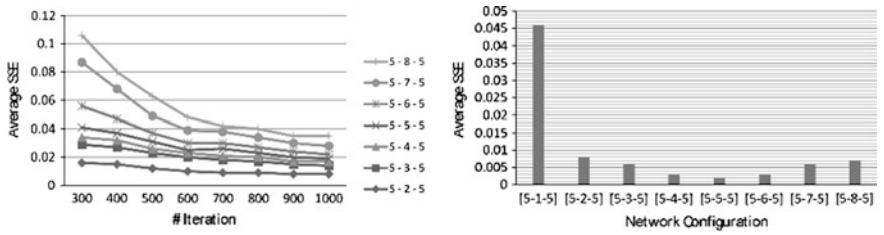

Fig. 7 (*left*) Convergence trajectory at NN Configuration; (*right*) SSE at various NN Configurations

Figure 7 demonstrates performance of BP algorithm based on network configuration, where horizontal axis (*X*-axis) indicates changes in number of nodes at hidden layer, while vertical axis (*Y*-axis) indicates average values of SSE obtained against various network configurations. From Fig. 7, it is evident that the algorithm performed best with a network configuration 5–4–5, where value 4 indicates number of nodes at hidden layer, whereas, for configuration higher than 5–5–5, performance of the algorithm was becoming poorer and poorer. Figure 7 illustrates BP algorithm convergence trajectory for its various network configurations. It may be observed that convergence speed was slower for higher network configurations than that of lower network configurations, and it is because of the fact that the higher network configurations have large number of free parameters in comparison to the lower network configurations. Therefore, the higher configurations need more number of iteration to converge than the lower network configuration.

We may increase number of hidden layers to form four- or five-layered NN. For the sake of simplicity, number of nodes was kept same at each hidden layer. It has been observed that computational complexity is directly proportional to number of hidden layers in network. It was obvious to bear additional computational cost if the performance of the algorithm improves for increasing number of hidden layers. It has also been observed that performances of algorithm and network configuration are highly sensitive to training set volume. Performance study based on training set volume is discussed in Sect. 3.1.6.

3.1.3 Initialization of Free Parameters (Initial Search Space—χ)

Proper choice of χ contributes to the performance of BP algorithm. Free parameters of BP algorithm were initialized with some initial guess. The remaining parameters were kept fixed at certain values, and synaptic weights were initialized between $[-0.5, 0.5]$ and $[-2.0, 2.0]$ in order to monitor influence of synaptic weights' initialization on performance of BP algorithm. Parameters setting was as follows: number of iteration was set to thousand, η to 0.5, β to 0.1, network configuration to 5–5–5 and training volume to fifty training examples.

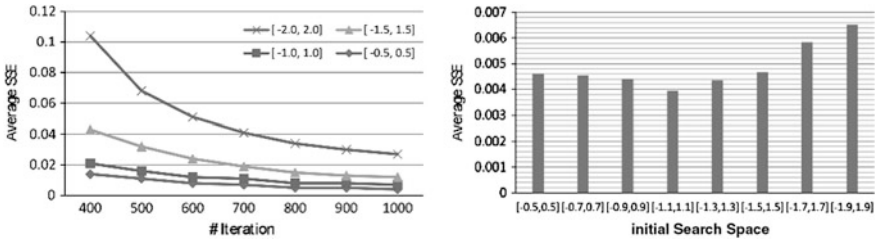

Fig. 8 (*left*) Convergence trajectory at χ; (*right*) SSE at various χ

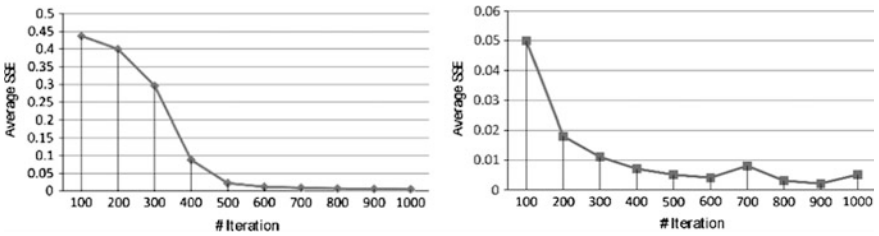

Fig. 9 (*left*) Convergence in batch mode; (*right*) Convergence in sequential mode

In Fig. 8, *X*-axis represents iteration, whereas, *Y*-axis represents average SSE values. The four continuous lines in Fig. 8 indicate convergence trajectory for various initialization ranges. Figure 8 demonstrates that large initial values lead to small local gradient that causes learning to be very slow. It was also observed that learning was good at somewhere between small and large initial values. In this case, ± 1.1 was a good choice of range χ.

3.1.4 Convergence Trajectory Analysis

In Fig. 9, *X*-axis indicates number of iteration, while *Y*-axis indicates average SSE achieved. Parameters setting was as follows: network configuration set at 5–5–5, number of iteration taken is 10,000, η taken is 0.5, β taken is 0.1 and training set volume is 50. Figure 9 indicates convergence trajectory of BP algorithm, where it may be observed that the performance of BP improves when iteration number was increased. For given training set and network configuration, average SSE gets reduced to 0.005 (SSE figures in this chapter are on normalized data) at iteration number 1000.

3.1.5 Learning Rate and Momentum Factor

Learning rate (η) is crucial for BP algorithm. It is evident from BP algorithm mentioned in Sect. 2.6.3 that weight changes in BP algorithm are proportional to derivative of error. Learning rate η controls weight changes. With larger η, larger

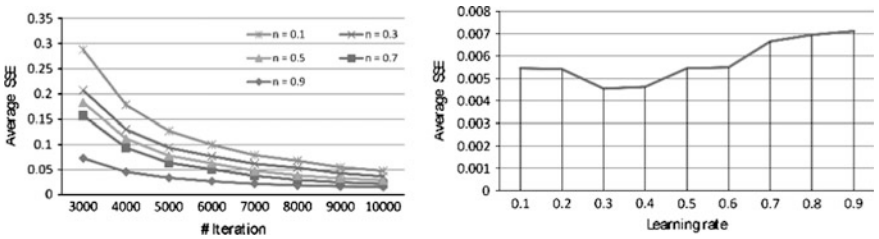

Fig. 10 (*left*) Convergence trajectories at η; (*right*) average SSE at various η

being weight changes in epochs. Training/Learning of NN is faster if η is larger and slower if η is lower. The size of η can influence networks ability to achieve a stable solution. A true gradient descent technique should take little steps to build solution. If η gets too large, then algorithm may disobey gradient descent procedure. Two distinct experiments have been done using η and their results are illustrated in Fig. 10.

In the first experiment, as shown in left of Fig. 10, network configuration was set to 5–5–5, number of iteration was set to 10,000, β was set to 0.1, training set volume was set to two-hundred and batch mode training was adopted. From this experiment, it was observed that for larger values of η, weight change ΔW_{ij} was larger and for smaller η, weight change ΔW_{ij} was smaller. At a fixed iteration 10,000 and at $\eta = 0.9$, network training speed was fast. We got SSE 0.02 at $\eta = 0.9$ and at $\eta = 0.1$, SSE was 0.05 which indicated that for small η, algorithm required more steps to converge, though convergence was guaranteed because small steps minimize the chance of escaping global minima. With larger values of η, algorithm may escape global minima that is clearly indicated in Fig. 10.

In the second experiment shown in right of Fig. 10, it may be observed that for lower η, i.e. 0.1 and 0.2, algorithm fell short to reach to a good result in limited number of iteration. However, at η 0.3 and 0.4, algorithm offered a good result, whereas, for learning rates higher than 0.4, algorithm tends to escape global optima.

For slight modification to BP weight update rule, additional momentum (β) term was introduced by Rumelhart [28]. Momentum factor was introduced to preserve weight change that should influence current search direction. The term momentum indicates that once weight starts moving in a particular direction, it should continue to move in that direction. Momentum can help in speeding-up learning and can help in escaping local minima. But too much speed may become unhealthy for a training process. Figure 11 indicates that for lower values of momentum rate, algorithm performs well. Therefore, β value should be increased as per learning speed requirements.

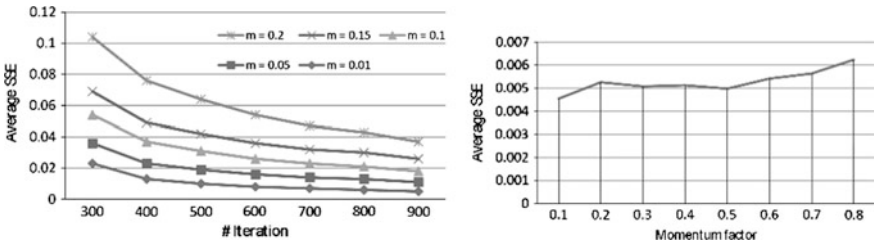

Fig. 11 (*left*) Convergence trajectories at β; (*right*) SSE at various β

3.1.6 Training Volume

We have already mentioned that performance of BP algorithm depends on its parameters and complexity of problem. We can regulate those parameters to improve the performance of the algorithm, but we cannot control the complexity of problem. Performance of an algorithm is highly influenced by the size of training set and the characteristic of data within training set. In Figs. 12 and 13, we present an observation based on volume of training set. In Fig. 12 (left), *X*-axis indicates volume of training set, whereas *Y*-axis indicates average SSE achieved. Figure 12 (left) indicates that average SSE has been plotted against different training set volumes feeding to networks of fixed size, trained at same iteration. From Fig. 12 (left), it may be observed that performance of the algorithm gets poorer on increasing volume of training set. To improve performance of the algorithm, we reconfigured NN to higher configuration and increased maximum training iteration values each time training set volume was increased. In Fig. 12 (right), first row values along *X*-axis indicate number of hidden layer nodes in a three-layered NN and second row values along *X*-axis indicate volume of training set, whereas *Y*-axis indicates average SSE achieved. In Fig. 13 (left), first row, second row and third row values along *X*-axis indicate maximum iteration, number of hidden layer nodes in three-layered NN and volume of training set, respectively, while *Y*-axis indicates average SSE achieved. From Fig. 12, it may be observed that increasing values of training set volume, average SSE was increased. Hence, the performance became

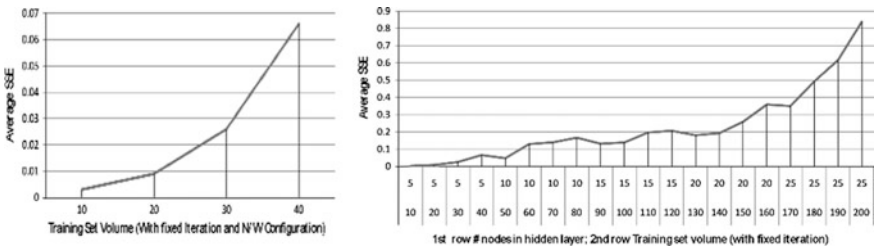

Fig. 12 (*left*) Fixed NN configuration and iteration (*right*) fixed iteration

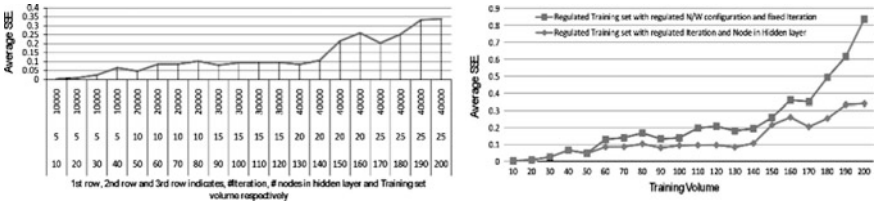

Fig. 13 (*left*) SSE versus training volume; (*right*) superimposition of Fig. 13 over Fig. 12

poorer, but as soon as network was reconfigured to a higher configuration, average SSE dips down. At that particular configuration, when training set volume was increased, SSE was also increased. Figure 13 (right) is a superimposition of Fig. 13 (left) over Fig. 12 (right). Figure 13 (right) indicates that iteration up-gradation and network reconfiguration together became necessary for improvement of algorithm when training set volume was increased.

3.2 Complexity Analysis

Time complexity of BP neural network algorithm depends on number of iteration it takes to converge, number of patterns in training data and time complexity needed to update synaptic weights. Hence, it is clear that time complexity of BP algorithm is problem dependent. Let n be the number of iterations required for convergence of BP algorithm and p be total number of patterns in training data. Synaptic weights of NN shown in Fig. 3 may be represented as a weight vector. Hence, to update synaptic weights, running time complexity required is $O(m)$, where m being the size of weight vector. To update synaptic weights, we need to compute gradient as per (5). The gradient computation for each pattern takes $O(n^2)$. NN weights may be updated either in sequential or batch mode as suggested in Sect. 3.1. In batch mode, weights are updated once in an epoch. One epoch training means updating NN weightes using entire trainning set at once. In sequential mode, weights are updated for each pattern presented to the network.

Let w be cost of gradient computation that is basically $O(m)$, equivalent to cost of updating weights. Whichever the training mode it may be, gradients are computed for each training pattern. In sequential mode, weight updating and gradient computation are parallel process. In batch mode, weight update once in an epoch, whose contribution is feeble in total cost and may be ignored. Hence, the complexity of BP algorithm stands to $O(p \times w \times n) = O(pwn)$.

3.3 Comparing BP with Hybrid NN Training Methods

We compared BP algorithm with three other intelligent techniques that were used for training of NN such as conjugate gradient (CG) method, genetic algorithm (GA) [30, 31] and particle swarm optimization (PSO) algorithm [32, 33]. This section provides a comprehensive performance study and comparison between these intelligent techniques that were applied for solving manhole gas detection problem.

3.3.1 Empirical Analysis

An empirical study is shown in Fig. 14, where *X*-axis indicates number of iterations while *Y*-axis indicates average SSE achieved against different iterations. Figure 14 (left) indicates convergence trajectory and Fig. 14 (right) indicates SSE obtained in various epochs in training process by each of the mentioned algorithms. It was observed that BP algorithm converged faster than the other algorithms. The convergence trajectory of CG method appeared smoother than that of BP algorithm, and its SSE value got reduced to value nearly equal to the value of SSE achieved by BP algorithm. Although the convergence trajectory of PSO approach was not as convincing as of BP algorithm and CG method, it was observed that PSO approach was efficient enough to ensure SSE nearer to the one achieved using classical NN training algorithms, BP and CG methods. Figure 14 indicates that GA was not as efficient as the other three approaches. GA quickly gets stuck into local optima as far as the present application was concerned.

3.3.2 Theoretical Analysis

The complexity analysis of BP algorithm is provided in Sect. 3.2. The cost met by line search in CG method was an additional computational cost in contrast with BP algorithm counterpart. The computational cost met by PSO and GA algorithm may be given as $O(pqwng)$, where q is size of population, n is number of iterations, p is

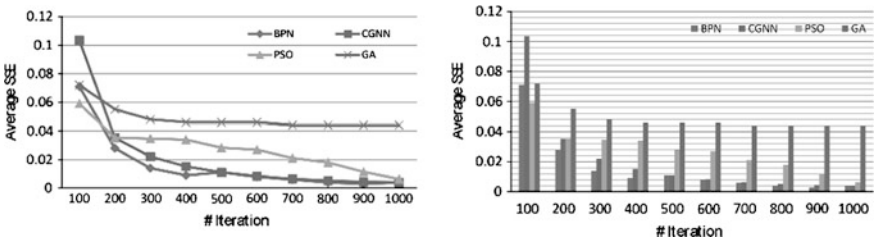

Fig. 14 (*left*) Convergence trajectory analysis; (*right*) SSE at various epochs

number of training examples, w is cost needed to update synaptic weights and g is cost of producing next generation. It may please be noted that the cost of g in PSO and GA depends on their own dynamics of producing next generation. In GA, it is based on selection, crossover and mutation operation [34], whereas PSO has simple non-derivative methods of producing next generation [35].

3.3.3 Statistical Analysis

We used Kolmogorov–Smirnov test (KS test), where KS test being nonparametric in nature, it does not make any assumption about data distribution. Two-sample KS test is useful for comparing two samples, as it is sensitive to differences in both location and shape of empirical cumulative distribution functions (*epcdf*) of two samples. Clearly speaking, KS test tries to determine if two datasets X and Y differ significantly [36]. KS test makes following hypothesis. Null hypothesis H_0 indicates that two underlying one-dimensional unknown probability distributions corresponding to X and Y are indistinguishable, i.e. datasets X and Y are statistically similar $(X \equiv Y)$. The alternative hypothesis H_t indicates that X and Y are distinguishable, i.e. datasets X and Y are statistically dissimilar. If it is alternative hypothesis, then their order (direction) becomes an important consideration. We need to determine whether the former (dataset X) is stochastically larger than or smaller than the latter one (dataset Y), i.e. $X > Y$ or $X < Y$. Such KS test is known as one-sided test that determines direction by computing distance D^{+}_{nm}, D^{-}_{nm}, and D_{nm}, where value n and m are cardinality of the set X and Y, respectively. The null hypothesis H_0 is rejected if $D_{nm}, > K\alpha$, where $K\alpha$ is known as critical value [37]. For n and m being 20, i.e. size of each sample, $K\alpha$ was found to be equal to 0.4301 for $\alpha = 0.05$ which indicates 95 % confidence in test. Readers may explore [37–39] to mitigate their more interest in KS test. KS test was conducted using twenty instances of SSEs produced by each mentioned algorithms. KS test was conducted in between BP and other intelligent algorithms. Hence, set X was prepared with SSEs of BP algorithm and three separate sets, and each representing set Y was prepared using SSE values of CG, PSO and GA algorithms. Outcome of KS test is provided in Table 3 where it is conclusive about significance of BP algorithm over the other algorithms.

Table 3 KS test: BP versus other intelligent algorithms

KS test type BP (X)	Intelligent algorithms (Y)		
	CGNN	PSO	GA
D^{+}_{nm}	0.10	0.15	1.00
D^{-}_{nm}	0.20	0.15	0.00
D_{nm}	0.20	0.15	1.00
Decision	$X \equiv Y$	$X \equiv Y$	$X > Y$

Table 4 System result presentation in ppm

Input gas	Responding unit			Safety limit (ppm)	Interpretation
	Sensor	NN	System (ppm)		
NH_3	0.260	0.016	80	25–40	Unsafe
CO	0.346	0.022	110	35–100	Unsafe
H_2S	0.240	0.023	115	50–100	Unsafe
CO_2	0.142	0.022	110	5000–8000	Safe
CH_4	0.843	0.993	4965	5000–10,000	Safe

4 Results and Discussion

Data sample was prepared as per the procedures mentioned in Sect. 2.5. The collected data sample was partitioned in two sets, training and test. The eighty percent of the original set was used as training set, and remaining twenty percent was used for testing purpose. After training, the system undergone a test using the test set. Output of the trained NN was de-normalized in order to present output in terms of concentration of gas components present in given test sample/gaseous mixture. We are providing a sample test result obtained for input sample 2, which is provided in Table 2. The predicted concentration value corresponding to the given input sample is shown in Table 4. In Table 4, interpretation column is based on comparison between de-normalized value of NN output and safety limits of the respective gases. Note that each of nodes at output layer is dedicated to a particular gas. Safety limits of manhole gases are as follows. Safety limit of NH_3 is in between 25 and 40 ppm (as per limit set by World Health Organization), CO is in between 35 and 100 ppm [40–42], H_2S is in between 50 and100 ppm [43, 44]), CO_2 is in between 5000 and 8000 ppm [41, 45] and CH_4 is in between 5000 and 10000 ppm [46].

5 Conclusion

In this chapter, we have discussed design issues of an intelligent sensory system (ISS) comprising semiconductor-based GSA and NN regressor. BP algorithm was used for supervised training of NN. The proposed design of ISS offered a solution to manhole gas mixture detection problem that was viewed as a noise reduction/pattern recognition problem. We have discussed mechanisms involved in preparation and collection of data sample for the development of proposed ISS. Cross sensitivity was firmly addressed in this chapter. We have discussed the issues in training of NN using backpropagation (BP) algorithm. A comprehensive performance study of BP algorithm was provided in this chapter. Performance comparison in terms of empirical, theoretical and statistical sense between BP and various other hybrid intelligent approaches applied on said problem was provided in

this chapter. A concise discussion on safety limits and systems' result presentation mechanism was presented in the remainder section of this chapter. Data sample mentioned in the scope of this research may not represent the entire spectrum of the mentioned manhole gas detection problem. Therefore, at present, it was a non-trivial task for NN regressor. Hence, it has offered a high-quality result. Therefore, an interesting study over larger dataset to examine how manhole gas detection problem can be framed as a classification problem using the available classifier tools.

Acknowledgments This work was supported by Department of Science & Technology (Govt. of India) for the financial supports vide Project No.: IDP/IND/02/2009 and the IPROCOM Marie Curie initial training network, funded through the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007–2013/ under REA grant agreement No. 316555.

References

1. Gromicko N (2006) Sewer gases in the home. http://www.nachi.org/sewer-gases-home.html. Accessed on 15 July 2015
2. Lewis (1996) Dangerous properties of industrial materials, 9th edn. Van Nostrand Reinhold, New York, ISBN 0132047674
3. Whorton J (2001) The insidious foe–sewer gas. West J Med 175(6):427428, (PMC 1275984)
4. Hindu T (2014a) Deaths in the drains. http://www.thehindu.com/opinion/op-ed/deaths-in-the-drains/article5868090.ece?homepage=true. Accessed on 15 July 2015
5. Hindu T (2014b) Sewer deaths. http://www.thehindu.com/opinion/letters/sewer-deaths/article5873493.ece. Accessed on 15 July 2015
6. Hindu T (2014c) Supreme court orders states to abolish manual scavenging. http://www.thehindu.com/news/national/supreme-court-orders-states-to-abolish-manual-scavenging/article5840086.ece. Accessed on 15 July 2015
7. NIOSH (2011) Volunteer fire fighter dies during attempted rescue of utility worker from a confined space. http://www.cdc.gov/niosh/fire/reports/face201031.html. Accessed on 15 July 2015
8. Time of India (2014) Sewer deaths http://timesofindia.indiatimes.com/city/delhi/Panel-holds-DDA-guilty-for-sewer-death/articleshow/31916051.cms. Accessed on 15 July 2015
9. Li J (1993) A mixed gas sensor system based on thin film saw sensor array and neural network. IEEE, pp 179–181, 0-7803-0976-6/93
10. Srivastava AK, Srivastava SK, Shukla KK (2000a) In search of a good neuro-genetic computational paradigm. IEEE, pp 497–502, 0-78O3-5812-0/00
11. Srivastava AK, Srivastava SK, Shukla KK (2000b) On the design issue of intelligent electronic nose system. IEEE, pp 243–248, 0-7803-5812-0/00
12. Llobet E (2001) Multicomponent gas mixture analysis using a single tin oxide sensor and dynamic pattern recognition. IEEE Sens J 1(3):207–213, ISSN 1530437X/01
13. Liu J, Zhang Y, Zhang Y, Cheng M (2001) Cross sensitivity reduction of gas sensors using genetic algorithm neural network. Farquharson S (ed) Proceedings of SPIE Optical Methods for Industrial Processes, vol 4201
14. Tsirigotis G, Berry L, Gatzioni M (2003) Neural network based recognition, of co and nh3 reducing gases, using a metallic oxide gas sensor array. In: Scientific proceedings of RTU, telecommunications and electronics, Series 7, vol. 3, pp 6–10

15. Lee DS, Ban SW, Lee M, Lee DD (2005) Micro gas sensor array with neural network for recognizing combustible leakage gases. IEEE Sens J 5(3):530–536. doi:10.1109/JSEN.2005.845186
16. Ambard M, Guo B, Martinez D, Bermak A (2008) A spiking neural network for gas discrimination using a tin oxide sensor array. In: International symposium on electronic design, test and applications, IEEE, pp 394–397, ISBN 0-7695-3110-5/08
17. Baha H, Dibi Z (2009) A novel neural network-based technique for smart gas sensors operating in a dynamic environment. Sensors 9:8944–8960, ISSN 1424–8220
18. Pan W, Li N, Liu P (2009) Application of electronic nose in gas mixture quantitative detection. In: Proceedings of IC-NIDC, IEEE, pp 976–980, ISBN 978-1-4244-4900-2/09
19. Wongchoosuka C, Wisitsoraatb A, Tuantranontb A, Kerdcharoena T (2010) Portable electronic nose based on carbon nanotube-sno2 gas sensors and its application for de- tection of methanol contamination in whiskeys. Sens Actuators B: Chem doi:10.1016/j.snb.2010.03.072
20. Zhang Q, Li H, Tang Z (2010) Knowledge-based genetic algorithms data fusion and its application in mine mixed-gas detection. IEEE, pp 1334–1338, 978-1-4244-5182-1/10
21. Won So D, Koo J (2010) The estimation of hazardous gas release rate using optical sensor and neural network. Pierucci S, Ferraris GB (eds) European Symposium on Computer Aided Process Engineering ESCAPE20, Elsevier B.V
22. Ojha VK, Duta P, Saha H, Ghosh S (2012c) Detection of proportion of different gas components present in manhole gas mixture using back-propagation neural network. In: International proceedings of computer science and information technology, vol 1, pp 11–15. ISBN 978-981-07-2068-1
23. Ojha VK, Duta P, Saha H, Ghosh S (2012) Linear regression based statistical approach for detecting proportion of component gases in man-hole gas mixture. In: International symposium on physics and technology of sensors, IEEE, doi:10.1109/ISPTS.2012.6260865
24. Ojha VK, Duta P, Saha H, Ghosh S (2012f) A novel neuro simulated annealing algorithm for detecting proportion of component gases in manhole gas mixture. International Conference on Advances in Computing and Communications, IEEE, pp 238–241, doi:10.1109/ICACC.2012.54
25. Ghosh S, Roy A, Singh S, Ojha VK, Dutta P, Saha H (2012a) Sensor array for manhole gas analysis. In: International Symposium on Physics and Technology of Sensors. IEEE, Pune, India, ISBN 978-1-4673-1040-6
26. Ghosh S, Roychaudhuri C, Saha H, Ojha VK, Dutta P (2012b) Portable sensor array system for intelligent recognizer of manhole gas. In: International conference on sensing technology (ICST), IEEE, India, pp 589–594, ISBN 978-1-4673-2246-1
27. Haykin S (2005) Neural networks a comprehensive foundation, 2nd edn. Pearson Prentice Hall, New Jersey, ISBN 81-7803-300-0
28. Rummelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533536. doi:10.1038/323533a0
29. Sivanadam SN, Deepa SN (2007) Principles of soft computing. Wiley, New Delhi, ISBN 81-265-1075-7
30. Ojha VK, Duta P, Saha H (2012) Performance analysis of neuro genetic algorithm applied on detecting proportion of components in manhole gas mixture. Int J Artif Intell Appl 3(4):83–98. doi:10.5121/IJAIA.2012.3406
31. Ojha VK, Duta P, Saha H, Ghosh S (2012b) Application of real valued neuro genetic algorithm in detection of components present in manhole gas mixture. Wyld DC (ed) Advances In Intelligent And Soft Computing, Springer, Vol. 166, pp 333–340, DOI: 10.1007/978-3-642-30157-5
32. Ojha VK, Duta P (2012) Performance analysis of neuro swarm optimization algorithm applied on detecting proportion of components in manhole gas mixture. Artif Intell Res 1:31–46. doi:10.5430/JNEP.V1N1PX
33. Ojha VK, Duta P, Saha H, Ghosh S (2012e) A neuro-swarm technique for the detection of proportion of components in manhole gas mixture. In: Proceedings of international conference on modeling, optimization and computing, vol 2. NI University, Kanyakumari, India, pp 1211–1218

34. Goldberg DE (2006) Genetic Algorithms in search, Optimization and machine learning, 1st edn. Pearson Education, Singapore, ISBN 81-7758-829-X
35. Kennedy J, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann Publishers, San Francisco, ISBN 1-55860-595-9
36. Dutta P, Dutta Majumder D (2012) Performance analysis of evolutionary algorithm. Lambert Academic Publishers, Saarbrücken, ISBN 978-3-659-18349-2
37. Gail MH, Green SB (1976) Critical values for the one-sided two-sample kolmogorov-Smirnov statistic. J Am Stat Assoc 71(355):757–760
38. Boes DC, Graybill FA, Mood AM (1974) Introduction to the theory of statistics, 3rd edn. McGraw-Hill, New York
39. Sheskin DJ (2003) Handbook of parametric and nonparametric statistical procedures, 3rd edn. CRC-Press, London, ISBN 1-58488-440-1
40. Ernst A, Zibrak JD (1998) Carbon monoxide poisoning. New Engl J Med 339(22) (PMID9828249)
41. Friedman D (2014) Toxicity of carbon dioxide gas exposure, CO_2 poisoning symptoms, carbon dioxide exposure limits, and links to toxic gas testing procedures. InspectAPedia http://www.inspectapedia.com/hazmat/CO2gashaz.htm#reviewers. Accessed on 15 July 2015
42. Goldstein M (2008) Carbon monoxide poisoning. J Emerg Nurs: JEN: Off Publ Emerg Dep Nurses Assoc (PMID 19022078)
43. USEPA (1980) Health and environmental effects problem for hydrogen sulfide. West J Med pp 118–8, eCAOCIN026A
44. Zenz C, Dickerson OB, EP Horvath (1994) Occupational medicine, 3rd edn. p 886
45. Shilpa G (2007) New insight into panic attacks: Carbon dioxide is the culprit. InspectAPedia
46. Fahey D (2002) Twenty questions and answers about the ozone layer