

# Application of Soft Computing Techniques to Multiphase Flow Measurement: A Review

Yong Yan<sup>1,2\*</sup>, Lijuan Wang<sup>2</sup>, Tao Wang<sup>3</sup>, Xue Wang<sup>4</sup>, Yonghui Hu<sup>1</sup>, Quansheng Duan<sup>1</sup>

<sup>1</sup> School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China

<sup>2</sup> School of Engineering and Digital Arts, University of Kent, Canterbury, Kent CT2 7NT, U.K.

<sup>3</sup> KROHNE Ltd., 34-38 Rutherford Drive, Wellingborough NN8 6AE, U.K.

<sup>4</sup> School of Mathematics, Statistics and Actuarial Science, University of Kent, Canterbury, Kent CT2 7NF, U.K.

\* corresponding author: y.yan@kent.ac.uk

## Abstract:

After extensive research and development over the past three decades, a range of techniques have been proposed and developed for online continuous measurement of multiphase flow. In recent years, with the rapid development of computer hardware and machine learning, soft computing techniques have been applied in many engineering disciplines, including indirect measurement of multiphase flow. This paper presents a comprehensive review of the soft computing techniques for multiphase flow metering with a particular focus on the measurement of individual phase flowrates and phase fractions. The paper describes the sensors used and the working principle, modelling and example applications of various soft computing techniques in addition to their merits and limitations. Trends and future developments of soft computing techniques in the field of multiphase flow measurement are also discussed.

**Keywords:** Multiphase flow measurement; Soft computing; Machine learning; Computational intelligence; Sensor fusion; Data-driven model

## Nomenclature

$a$	Bias
$b$	Bias
$C$	User-specified parameter
$C_b$	Concentration of biomass
$C_c$	Concentration of coal
$C_j$	The centre vector for the $j^{th}$ hidden node
$d$	A constant
$f(x)$	Transfer function
$H_i$	The $i^{th}$ hidden neuron
$L$	Number of hidden nodes
$m$	Number of input variables
$n$	Number of training samples
$O$	Node in the adaptive neuro-fuzzy inference system
$q_m$	Mass flow rate of the mixture
$q_{mg}$	Mass flow rate of gas flow

$q_{ml}$	Mass flow rate of liquid flow
$q_{mo}$	Mass flow rate of oil flow
$q_{ms}$	Mass flow rate of solid flow
$q_{mw}$	Mass flow rate of water flow
$q_{v,g}$	Volumatic flow rate of gas flow
$q_{v,l}$	Volumatic flow rate of liquid flow
$t_i$	The vector of outputs from the $i^{th}$ tree
$v$	Velocity of the mixture
$v_g$	Velocity of gas flow
$v_l$	Velocity of liquid flow
$v_o$	Velocity of oil flow
$v_s$	Velocity of solid flow
$x$	Input vector
$x_i$	The $i^{th}$ element in the input vector $\mathbf{x}$
$X^*$	The matrix of training samples
$y$	The desired output
$Y$	The vector of desired output
$\alpha_i$	The $i^{th}$ Lagrange multiplier
$\varepsilon$	The approximation accuracy
$\theta$	A constant
$\mu(x)$	A membership function
$\xi_i$	A slack variable
$\xi_i^*$	A slack variable
$\sigma^2$	The variance of the Gaussian function
$\Phi(x)$	Transfer function
$\phi$	A constant
$\omega_i$	The $i^{th}$ weight
$\omega$	The vector of weights

## 1. Introduction

Multiphase flow is defined as a simultaneous flow of materials with two or more different phases (i.e. gas, liquid or solid) or unseparated components (e.g. water and oil) [1]. Multiphase flow (including two-phase flow which is a common example of multiphase flow) is widely seen in many industrial processes. Oil/gas/water mixtures are perhaps the most common gas-liquid and liquid-liquid two-phase or three-phase flows during the processes of production, transportation and custody transfer in the oil and gas industry. Meanwhile, air entrainment is unavoidable when marine fuel is transferred from a bunker barge to a receiving ship, particularly during the start and stop processes of the bunkering. For the fiscal purpose, accurate mass flow metering of marine fuel is essential in bunkering centres. Pneumatically

conveyed pulverized fuel (coal, biomass or mixture of both) in power plants forms gas-solid two-phase or three-phase flow. Individual flowrates and fractions of biomass and pulverized coal provide the information of co-firing ratio, which is useful to improve the combustion efficiency and reduce emissions of  $\text{NO}_x$  and  $\text{CO}_x$ . In some circumstances hydraulic transport of solids, such as sand, iron concentrates and phosphate matrix, in the type of slurry flow is employed in the mining, chemical, pharmaceutical and food industries. In such industrial processes accurate measurement of multiphase flow is highly desirable to realize flow quantification, operation monitoring, process optimization, and product quality control. It must be noted that multiphase flow is not restricted in industrial processes and covers many other application areas such as regional particle deposition and airflow in human tracheobronchial airways [2, 3] in the medical area. However, this review focuses primarily on the measurement of multiphase flow in the process and related industries.

Individual flowrates (volumetric flowrate or mass flowrate) and phase fractions are most important parameters to characterize a multiphase flow. Over the past three decades substantial progress has been made to develop new techniques that may offer solutions to the industrial measurement challenges. Thorn et al. [4,5] and Falcone et al. [6] have reviewed the developments of three-phase flowmeters, particularly for the petroleum industry. Possible techniques for the measurement of gas-solid flow in pneumatic conveying pipelines and circulating fluidized beds have been discussed in detail by Yan [7], Zheng and Liu [8], and Sun and Yan [9]. Albion et al. [10] have reviewed the intrusive and non-intrusive measurement techniques for monitoring slurry transportation in horizontal pipelines. Among these techniques, on-line multiphase flowmeters are the devices to measure the mixed flow without any separators and sampling lines. They can be classified into direct and indirect measurement groups according to measurement strategies deployed. The direct measurement of a phase flowrate is often realized using a Venturi flowmeter, Coriolis flowmeter and cross-correlation techniques etc., whilst a phase fraction is usually determined from radiation absorption, electrical impedance and microwave techniques etc. An indirect measurement method determines the individual phases through the analysis of the time variant signals acquired from a set of sensors. In general, the relationship between the sensor outputs and the flowrate or fraction of each phase cannot be deduced theoretically. In this case, empirical models are commonly developed from experimental data using statistical methods. With the recent development of artificial intelligence and machine learning, soft computing techniques provide alternative approaches to traditional statistical methods and extend the capabilities of empirical models.

This review focuses on the indirect methods incorporating soft computing techniques to measure the individual phase flowrates and fractions of multiphase flow. Section 2 outlines the principal constituents of soft computing techniques and provides a brief description of some techniques which are already applied in multiphase flow measurement, i.e. artificial neural

network (ANN), support vector machine (SVM), genetic algorithm (GA), genetic programming (GP) and adaptive neuro-fuzzy system (ANFIS). Section 3 presents the example applications of soft computing techniques in two-phase or three-phase flow measurement. Section 4 summarizes the findings of the review and discusses the trends and future developments of soft computing techniques in the field of multiphase flow measurement. Section 5 concludes this review and likely future development.

## **2. Soft Computing Techniques**

Soft computing is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solution cost [11]. It is sometimes referred as computational intelligence, covering a range of computational techniques in computer science, artificial intelligence and machine learning. Sometimes, the term ‘soft computing’ is used interchangeably with soft sensors or virtual sensors. Soft sensor is a common name for a piece of software which is used to derive desirable information from available measurements. Soft sensors are especially useful in data fusion, where measurements of different characteristics and dynamics are fused and combined. Well-known software algorithms that are used in soft sensors include Kalman filters [12]. Recently neural networks or fuzzy computing have been used to implement soft sensors. To some extent, the software developed based on soft computing techniques for the purpose of monitoring or measurement can be regarded as a soft sensor.

As shown in Fig.1, the principal constituent techniques of soft computing include machine learning (neural network, support vector machine and deep learning, etc.), evolutionary computation (evolutionary programming, genetic algorithm, evolution strategy and genetic programming etc.), fuzzy logic and probabilistic reasoning (Bayesian belief net and Dempster-Shafer theory, etc.). Machine learning and evolutionary computation are data-driven search and optimization approaches while fuzzy logic and probabilistic reasoning techniques are based on knowledge-driven reasoning. Each technique can be used independently whilst a combination of several techniques constitutes hybrid models.

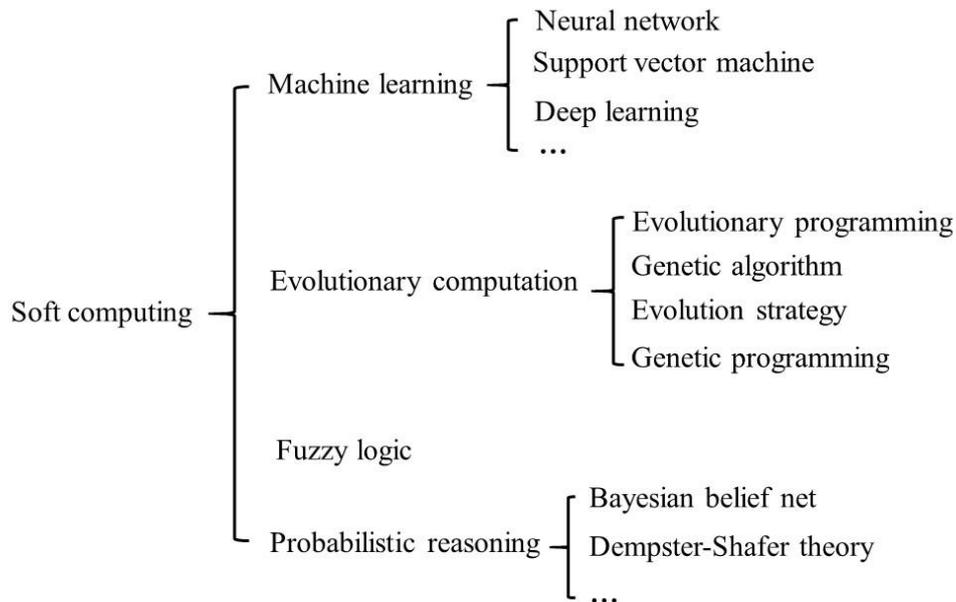


Fig.1 Principal constituents of soft computing techniques

In recent years soft computing has become a promising tool in resolving engineering challenges due to its ability to handle highly complex, dynamic and non-linear problems and computational simplicity over analytical methods. For these reasons, soft computing techniques have been widely used in many fields, in particular, computer engineering, environmental engineering, material engineering and medical diagnosis [13]. In the case of multiphase flow metering, soft computing techniques are used to extract useful information from sensor outputs to predict or estimate the flow rates and fractions of multiphase flow or identify flow patterns.

## 2.1 Artificial Neural Network

ANN models are developed by training a network to represent the relationships or processes that are inherent within the data. Being essentially non-linear regression models, they perform an input-output mapping using a set of interconnected nodes or neurons. Each neuron takes in inputs either externally or from other neurons and passes them on through an activation or transfer function.

In general, ANNs have three kinds of topology: multilayer, single-layer and recurrent. A multilayer ANN normally consists of an input layer, one or more hidden layers and an output layer. For a single-layer ANN, there is no hidden layer. A recurrent ANN includes at least a feedback loop in the network. Another way to classify ANNs is dependent on the learning strategy: supervised or unsupervised. In supervised learning, the training data set consists of both input objects and desired outputs while in the unsupervised learning the data are unlabelled. A variety of ANN models have been developed and used for a range of applications

such as pattern recognition, associative memory, optimization, function approximation, modelling and control, image processing and classification [14].

Table 1 summarizes the topology, learning strategy and function of the commonly used ANNs in engineering, including MLP (Multi-Layer Perception), RBF (Radial Basis Function), WNN (Wavelet Neural Network), ELM (Extreme Learning Machine), Elman, Hopfield, Kohonen, PNN (Probabilistic Neural Network), CNN (Cellular Neural Network), ART (Adaptive Resonance Theory), CMAC (Cerebellar Model Articulation Controller) and CM (Committee Machine) [15,16].

Table 1 Topology, learning strategy and function of the commonly used ANNs

ANN	Topology	Learning strategy	Function
MLP	multilayer feedforward	supervised	Pattern recognition, function approximation, modelling and control, classification
RBF	three-layer feedforward	supervised	Function approximation, classification
WNN	three-layer feedforward	supervised	Forecast, classification, function approximation
ELM	three-layer feedforward	supervised	Function approximation, classification
Elman	recurrent	supervised	Time-series forecast, pattern recognition
Hopfield	recurrent	supervised	Pattern recognition, associative memory, optimization, image processing
Kohonen	single layer	unsupervised	Pattern recognition, associative memory, classification
PNN	four-layered feedforward	supervised	Pattern recognition, classification
CNN	multilayer	supervised	Optimization, classification
ART	recurrent	unsupervised	Optimization, classification
CMAC	multilayer	supervised	Function approximation, modelling and control
CM	multiple NNs (experts)	supervised / unsupervised	Pattern recognition, function approximation, modelling and control, classification

As MLP and RBF neural networks have been applied to multiphase flow measurement, a brief introduction to these two kinds of ANN is given in this section.

### 2.1.1 MLP

Fig.2 depicts a typical  $m$ -input-1-output three-layer ANN.  $\mathbf{x}=(x_1, x_2, \dots, x_m)^T$  is an input sample and  $y$  is the desired output. Assume  $y$  is the linear output of the hidden neurons and a transfer function  $f(x)$  is used on the neurons, the ANN is modelled as [16]:

$$y = \sum_{j=1}^L \omega_j H_j + b = \sum_{j=1}^L \omega_j f\left(\sum_{i=1}^m \omega_{ij} x_i + a_j\right) + b \quad (1)$$

where  $m$  and  $L$  are the numbers of input variables and hidden nodes, respectively.  $\omega_j$  is the weight connecting the  $j^{\text{th}}$  hidden node and the output node,  $\omega_{ij}$  is the weight connecting the  $i^{\text{th}}$  input node to the  $j^{\text{th}}$  hidden node.  $a_j$  and  $b$  are the biases on the  $j^{\text{th}}$  hidden node and the output node, respectively. The weights ( $\omega_{ij}$  and  $\omega_j$ ) and biases ( $a_j$  and  $b$ ) between the layers are obtained through a training process.

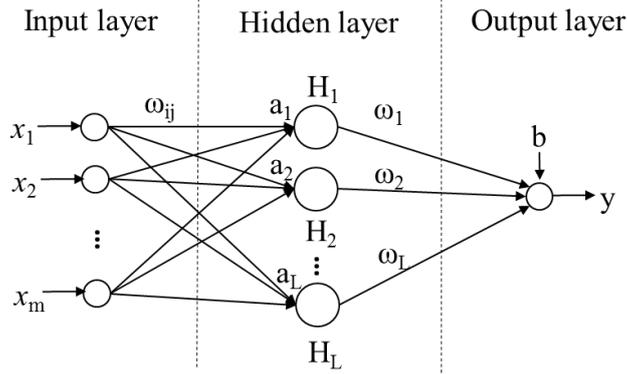


Fig.2 Structure of an  $m$ -input-1-output three-layer ANN

The performance of a neural network depends on the choice of its structure and learning algorithm. Design considerations of the ANN structure include the number of inputs, outputs, hidden layers and hidden neurons. The network inputs can be any combination of variables that are thought to be significant for predicting the output. Therefore, some knowledge of the problem and the procedures of input variable selection are very important. The number of hidden layers is determined according to the complexity of the problem. In general, the more complex the problem, the more hidden layers are required to achieve a good approximation level. Meanwhile, a trade-off between the approximation level and computational cost should be compromised. The number of neurons ( $L$ ) in the hidden layer can be determined using some guidelines [17] such as

$$L \leq 2m + 1 \quad (2)$$

$$L \leq \frac{n}{m+1} \quad (3)$$

where  $m$  and  $n$  are the numbers of input variables and training samples, respectively. However, these rules only give the range of  $L$ . The exact  $L$  for the model should be selected through trial-and-error to compromise between minimizing errors and achieving good generalization capability. The number of nodes in the output layer is equal to the number of desired output variables.

The transfer function between the layers can be sigmoid, hyperbolic tangent sigmoid, linear, Gaussian and exponential etc., which is determined according to the performance of the network. Backpropagation (BP) algorithm is one of the most popular and robust tools in the training process. It is a variation of the gradient descent optimisation algorithm that minimises the error between the predicted and actual output values. The weights between neurons are adjusted after each training cycle until the error in the validation data set begins to rise.

### 2.1.2 RBF

As shown in Fig.3, RBF-ANN has a fixed three-layer structure and applies a type of radial basis function as an activation function to the hidden nodes. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters. The radial basis function measures the distance between the input vectors and weight vectors and is typically taken to be the Gaussian function. The output of the network is given by [16]

$$y = \sum_{j=1}^L \omega_j H_j = \sum_{j=1}^L \omega_j \exp\left(-\frac{1}{2\sigma^2} \|x - C_j\|^2\right) \quad (4)$$

where  $C_j$  is the centre vector for the  $j^{\text{th}}$  hidden node and determined by the K-means clustering method.  $\|x - C_j\|$  is the Euclidean norm and  $\sigma^2$  is the variance of the Gaussian function.

An RBF network with enough hidden nodes can approximate any continuous function with arbitrary precision. Moreover, as a local approximation network, the RBF neural network has the advantages of simple structure, less adjective parameters and efficient training.

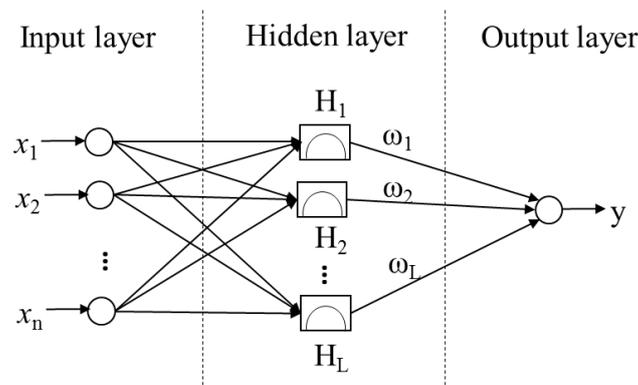


Fig.3 Structure of an RBF-ANN.

## 2.2 Support Vector Machine

SVM was developed by Vapnik in 1995 to solve the classification problem based on the statistic learning theory and structural risk minimization [18]. Since then, this method has been extended to the domain of regression and prediction problems [19]. As shown in Fig.4, SVM maps the data  $\mathbf{x}$  from the input space to a feature space through the nonlinear mapping  $\varphi(\mathbf{x})$ . Constrained-optimization methods are used to identify the separating hyperplane which maximizes the separating margins of two different classes in the feature space [18].

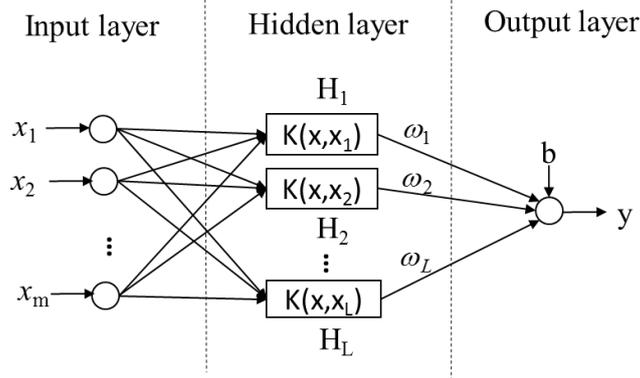


Fig.4 Structure of an SVM

### 2.2.1 SVM for Classification

Given  $n$  training samples  $X^*=(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$  and the corresponding desired output  $Y=(y_1, y_2, \dots, y_n)$ , each input sample is a vector  $\mathbf{x}=(x_1, x_2, \dots, x_m)^T$  consisting of  $m$  variables. The input vectors are mapped into an  $L$ -dimensional feature space using a transfer function  $\varphi(\mathbf{x})$  [17].

The distance between two different classes in the feature space is  $\frac{2}{\|\omega\|}$ . To maximize the separating margin and to minimize the training errors  $\xi_i$  is equivalent to [20]

$$\begin{aligned} \min & \frac{1}{2}\|\omega\|^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} & \begin{cases} y_i \langle \omega, \varphi(\mathbf{x}_i) \rangle + b \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \end{aligned} \quad (5)$$

where  $\xi_i$  is the slack variable and  $C$  is a user-specified parameter and provides a trade-off between the distance of the separating margin and the training error.

Based on the Karush-Kuhn-Tucker (KKT) theorem, the problem is equivalent to solve the following dual problem:

$$\begin{aligned} \min & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle - \sum_{i=1}^n \alpha_i \\ \text{s.t.} & \begin{cases} \sum_{i=1}^n y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases} \end{aligned} \quad (6)$$

where each Lagrange multiplier  $\alpha_i$  corresponds to a training sample  $(\mathbf{x}_i, y_i)$ .  $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle$  is a kernel function. The decision function of the SVM classifier is described as

$$y = f(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \right) \quad (7)$$

There are some optional kernel functions for SVM, for example:

$$\text{Linear kernel: } K(\mathbf{x}, \mathbf{x}_i) = \langle \mathbf{x}, \mathbf{x}_i \rangle \quad (8)$$

$$\text{Polynomial kernel: } K(\mathbf{x}, \mathbf{x}_i) = (\langle \mathbf{x}, \mathbf{x}_i \rangle + p)^d \quad d \in m, p > 0 \quad (9)$$

$$\text{RBF kernel: } K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right)$$

(10)

$$\text{Sigmoid kernel: } K(\mathbf{x}, \mathbf{x}_i) = \tanh(\phi \langle \mathbf{x}, \mathbf{x}_i \rangle + \theta) \quad \phi > 0, \theta > 0 \quad (11)$$

### 2.2.2 SVM for Regression

SVM regression performs a linear regression in the high dimensional feature space using  $\varepsilon$ -insensitive loss and tends to reduce the model complexity by minimizing  $\|\omega\|^2$  [17]. This can be described by introducing slack variables  $\xi_i$  and  $\xi_i^*$  ( $i=1,2,\dots,n$ ) to measure the deviation of training samples ( $X^*, Y$ ) outside  $\varepsilon$ -insensitive zone. Regression estimates can be obtained by minimizing the empirical risk on the training data and thus the optimization problem is formulated as [20]

$$\begin{aligned} \min & \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ \text{s.t.} & \begin{cases} y_i - \langle \omega, \varphi(\mathbf{x}_i) \rangle - b \leq \varepsilon + \xi_i \\ \langle \omega, \varphi(\mathbf{x}_i) \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (12)$$

where  $n$  is the number of training samples,  $\varepsilon$  is the approximation accuracy that can be violated by means of the slack variables  $\xi$  and  $\xi^*$  for the non-feasible case.  $C$  is a positive constant as a regularization parameter that allows tuning the trade-off between the flatness of the function  $f(\mathbf{x})$  and the tolerance of deviations larger than  $\varepsilon$ . The parameter  $b$  can be computed by exploiting the Karush-Kuhn-Tucker (KKT) conditions.

The problem is transferred into a dual problem by the Lagrangian function:

$$\begin{aligned} \max_{\alpha, \alpha^*} Q &= -\frac{1}{2} \sum_{i,k=1}^n (\alpha_i - \alpha_i^*) (\alpha_k - \alpha_k^*) \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_k) \rangle - \varepsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \\ \text{s.t.} & \begin{cases} \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned} \quad (13)$$

where  $\alpha_i, \alpha_i^* \geq 0$  are Lagrange multipliers. According to Mercer's condition, the inner product  $\langle \varphi(\mathbf{x}), \varphi(\mathbf{x}_i) \rangle$  can be defined through a kernel  $K(\mathbf{x}, \mathbf{x}_i)$ , so the final product of a training process in the SVM method can be presented by:

$$f(\mathbf{x}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(\mathbf{x}, \mathbf{x}_i) + b \quad (14)$$

Support vector machines are supervised learning models with associated learning algorithms that analyse data for classification and regression analysis [21]. The original formulation is presented with a set of labelled data instances and the SVM training algorithm aims to find a hyperplane that separates the dataset into a discrete predefined number of classes in a fashion consistent with the training examples. In recent years, some optimized models of SVM, such as least squares SVM [22], twin SVM [23], multi-kernel SVM [24] and fuzzy SVM [25], have been developed to improve the computational efficiency and generalization ability. They have been successfully applied to a variety of real-world problems like particle identification, facial recognition, text categorization, bioinformatics, civil engineering and electrical engineering [26].

## 2.3 Evolutionary Computation

Evolutionary computation is inspired by the principles of genetics and natural selection. The core constituents of evolutionary computation or evolutionary algorithms include genetic algorithms (GA), evolution strategies (ES), evolutionary programming (EP) and genetic programming (GP) [27]. These evolutionary approaches provide alternative solutions to the problems where heuristic solutions are not available or unsatisfactory. Due to the advantages of simplicity, flexibility and robustness, evolutionary algorithms have been successfully applied to various problems in different areas, including optimization, automatic programming, machine learning, operations research, bioinformatics and social systems etc. GA and GP are two of the most representative approaches in evolutionary computation.

### 2.3.1 GA

Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by applying bioinspired operators such as mutation, crossover and selection. As shown in Fig.5, the evolution usually starts from a population of randomly generated individuals. In each generation, the fitness of every individual in the population is evaluated with an objective function. The fitter individuals are stochastically selected from the current population and each individual's genome is modified to form a new generation. The new generation of candidate solutions is used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached.

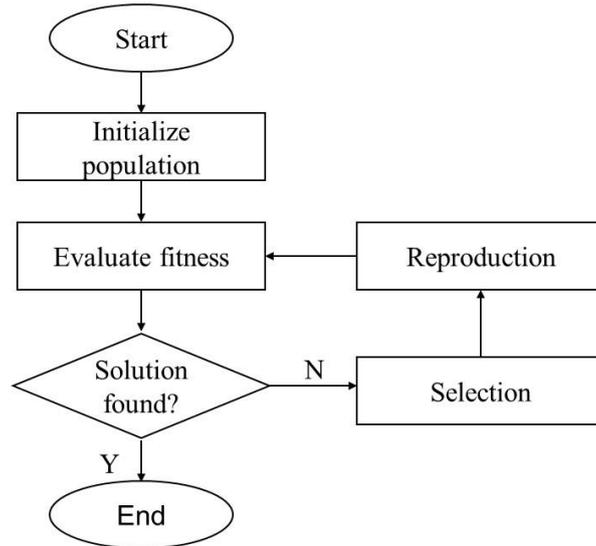


Fig.5 Flowchart of GA algorithm

### 2.3.2 GP

Genetic programming is a method for evolving equations by taking various mathematical building blocks such as functions, constants and arithmetic operations and combining them into a single expression. It was originally developed by Koza in 1992 [28]. The main difference between genetic programming and genetic algorithm is the representation of the solution. GP as an advanced evolutionary computation technique is an extension of genetic algorithms and is widely applied to symbolic data mining (symbolic regression, classification and optimization) [28,29]. Unlike traditional regression analysis, GP based symbolic regression automatically evolves both the structure and parameters of the mathematical model from the available data. Meanwhile, it is superior to other machine learning techniques in terms of the ability to generate an empirical mathematical equation without assuming prior form of the existing relationships. Fig.6 shows the structure of a multigene symbolic regression model.

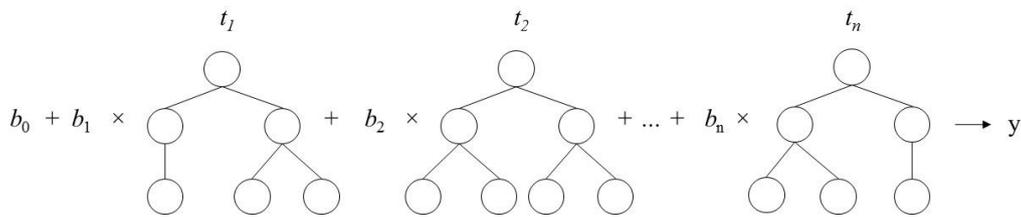


Fig.6 Structure of a GP model

The GP model can be regarded as a linear combination of lower-order nonlinear transformations of the input variables. The output  $y_{GP}$  is defined as a vector output of  $n$  trees modified by the bias term  $b_0$  and scaling parameters  $b_1, \dots, b_n$ :

$$y_{GP} = b_0 + b_1 t_1 + \dots + b_n t_n \quad (15)$$

where  $t_i$  ( $i=1, \dots, n$ ) is the  $(m \times 1)$  vector of outputs from the  $i^{th}$  tree comprising a multigene individual. The evolutionary process starts with initial population by creating individuals containing GP trees with different genes generated randomly. The evolutionary process

continues with an evaluation of the fitness of the new population, two-point high-level crossover to acquire and delete genes and low-level crossover on sub-trees. Then the created trees replace the parent trees or the unaltered individual in the next generation through mutation operators. The best program that appeared in any generation, the best-so-far solution, defines the output of the GP algorithm.

## **2.4 Fuzzy Logic**

The theory of fuzzy logic was formulated by Zadeh in 1965 [30]. Fuzzy logic provides inference mechanism that enables approximation reasoning, models human reasoning capabilities in the knowledge-based systems and deals with imprecision and uncertainty. It consists of fuzzy sets, membership functions and rules set for solving various computational problems. In the fuzzy system theory, an element can belong to a set with a certain degree (partial membership). The degree of membership is referred to as the membership value, and is commonly represented by a real value in  $[0,1]$ . Fuzzy set, therefore, provides a powerful computational paradigm for extending the capability of binary logic in ways that enable a much better representation of knowledge in a specific application.

Fuzzy logic has been applied in diverse areas, such as control systems, pattern recognition, forecasting, reliability engineering, signal processing, monitoring and medical diagnosis [31].

## **2.5 Probabilistic Reasoning**

A Bayesian network (belief network) is a probabilistic graphical model that represents a set of variables and their probabilistic dependencies. Formally, Bayesian networks are directed acyclic graphs whose nodes represent random variables in the Bayesian sense: observed quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies. Unconnected nodes represent variables that are conditionally independent of each other. Each node is associated with a probability function that takes a particular set of values for the node's parent variables and gives the probability of the variable represented by the node.

Bayesian networks are used for modelling beliefs in computational biology and bioinformatics medicine, biomonitoring, document classification, information retrieval, semantic search, image processing, data fusion, decision support systems, financial and marketing informatics, and risk analysis [32].

## **2.6 Hybrid Models**

Hybrid models integrate two or more soft computing techniques to solve a problem, such as neural network combined with GA (neuro-genetic) [33], neural network combined with fuzzy logic (neuro-fuzzy) [34], fuzzy logic system incorporating GA (fuzzy-genetic) [35] and neural

network incorporating fuzzy logic and GA (neuro-fuzzy-genetic) [36]. In a neuro-genetic model, the neural network calls a genetic algorithm to optimize its structural parameters and hence a model with better performance. In a neuro-fuzzy model, ANN develops acceptable if-then rules and membership functions for a fuzzy logic system from the given input-output information pairs. The hybrid system takes advantage of both neural network and fuzzy logic system. The integration of soft computing techniques provides complementary reasoning and searching methods, which are combined with domain knowledge and empirical data to develop flexible computing tools and solve complex problems. Hybrid soft computing models have been applied to a large number of classification, prediction and control problems and have the potential to act as more efficient and intelligent models.

A typical hybrid model is adaptive neuro-fuzzy inference system (ANFIS), in which a fuzzy inference system is implemented in the framework of adaptive networks. It has the advantages of the learning property of ANN and the expert knowledge of the fuzzy inference system [37,38]. The ANFIS model is a multilayer neural network-based fuzzy system which has a total of five layers. The structure of ANFIS with two inputs is illustrated in Fig.7. The input and output nodes represent the input states and output response, respectively. In the middle layers, there are nodes functioning as membership functions and rules.

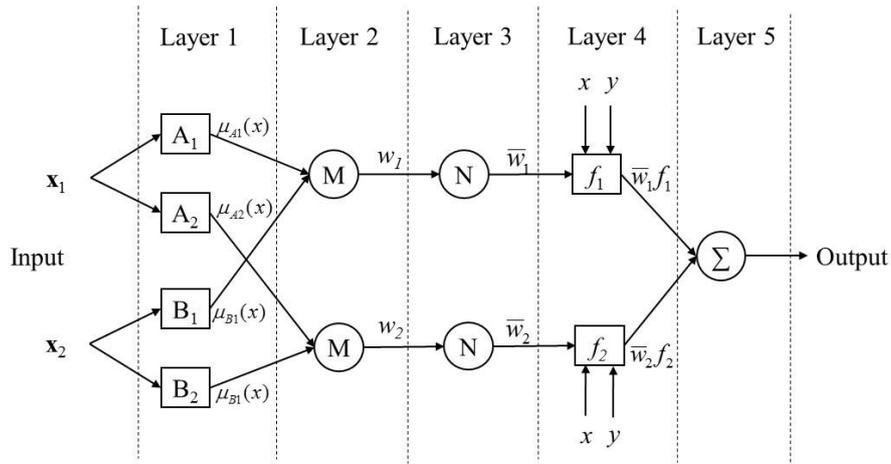


Fig.7 Structure of the ANFIS model

As shown in Fig.7, the node  $O_{1,i}$  in layer 1 is an adaptive node with a membership function

$\mu(x)$ :

$$\mu(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \quad (16)$$

$$O_{1,i} = \mu_{A_i}(x_1) \quad , i=1, 2 \quad (17)$$

$$O_{1,i} = \mu_{B_{i-2}}(x_2) \quad , i=3, 4 \quad (18)$$

where  $\mu_{A_i}$  and  $\mu_{B_{i-2}}$  are the degree of membership functions for the fuzzy sets  $A_i$  and  $B_i$ , respectively.  $\{a_i, b_i, c_i\}$  is the parameter set, referred as the premise parameters.

The node  $O_{2,i}$  in layer 2 multiply the incoming signals and send the product out. Each node output represents the firing strength of a rule.

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i=1, 2 \quad (19)$$

The node  $O_{3,i}$  in layer 3 is a fixed node and calculate the normalized firing strengths  $\bar{w}_i$ .

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i=1, 2 \quad (20)$$

The node  $O_{4,i}$  in layer 4 is an adaptive node with a node function and compute the contribution of the  $i^{th}$  rule towards the model output:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \quad i=1, 2 \quad (21)$$

where  $\bar{w}_i$  is the output of layer 3 and  $\{p_i, q_i, r_i\}$  is the parameter set, referred to as the consequent parameters.

The single node  $O_5$  in layer 5 is a fixed node and computes the overall output as the summation of all incoming signals:

$$O_5 = \sum_{i=1}^2 \bar{w}_i f_i = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} \quad (22)$$

### 3. Applications of Soft Computing Techniques

The applications of soft computing techniques to multiphase flow measurement are mainly concentrated on the estimation of phase flowrates and phase fractions and the identification of flow regime. The estimation of phase flowrates and phase fractions is equivalent to solve a problem of function approximation while the identification of flow regime is a classification problem. As the review focuses on the measurement of phase flowrates and phase fractions, the research purely on flow regime identification using soft computing techniques is out of the scope of this review. The following section reviews indirect multiphase measurement systems combining traditional sensors and soft computing techniques.

#### 3.1 Ultrasonic Sensors

Figueiredo et al. [39] made use of four ultrasonic sensors incorporating an ANN to identify the flow pattern and obtain the gas volume fraction of two-phase flow, respectively. It can be seen from Fig.8 that the input of the ANN is comprised of energy ratios from four acoustic sensors. There are two hidden layers, including five and two hidden neurons respectively in the ANN model. The output layer generates the identified flow pattern or the estimated gas volume fraction. Experimental work with air-oil flow was conducted on 1-inch and 2-inch vertical test sections, respectively. During the experimental tests, dispersed bubbles, intermittent flow,

churn flow and annular flow were observed with the liquid velocity ranging from 0.1 to 0.3 m/s and the gas volume fraction from 0 to 85%. Experimental results showed that the overall successful recognition rate of flow pattern was 98.3% while the overall variation of the estimated gas volume fraction was  $\pm 4.2$ .

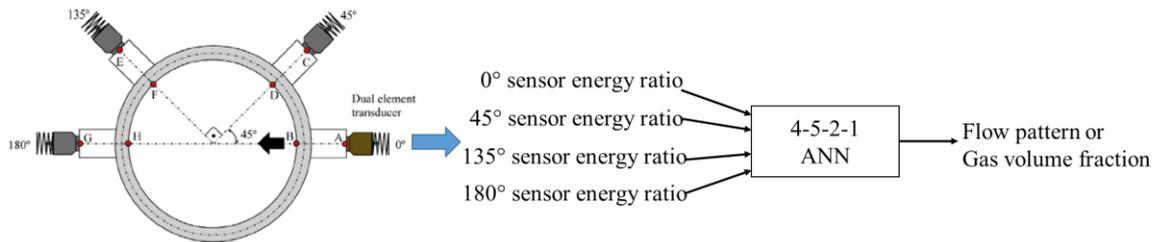


Fig.8 Measurement system based on ultrasonic sensors and ANN [39]

### 3.2 Differential Pressure Devices

Xu et al. [40] proposed a novel approach to the measurement of wet gas flow using a throat-extended Venturi meter and soft computing approximation techniques. The measurement system is shown in Fig.9. A backpropagation ANN (BP-ANN) model and a SVM model were developed to estimate the gas flowrate and liquid flowrate through the static and dynamic features of differential pressures. Experimental tests were carried out with natural gas and water two-phase flow on a 2-inch test rig. The gas flowrate was between 0.0139 and 0.0444 m<sup>3</sup>/s and liquid flowrate ranging from  $3.0556 \times 10^{-4}$  to 0.0015 m<sup>3</sup>/s. It was found that both ANN and SVM models were valid in the approximation of the complex relationship between the signal features and the two-phase flowrates. With the usage of the BP-ANN, the mean prediction error and standard deviation were 3.14% and 4.22% for gas flowrate, respectively, whereas the mean and standard deviation were 4.77% and 6.33% for water flowrate, respectively. Through the SVM model, the mean and standard deviation of the relative prediction errors were 2.86% and 4.39%, respectively, for gas flowrate, whereas the mean and standard deviation were 4.25% and 6.09%, respectively, for water flowrate. In Comparison with the ANN model, the SVM model was clearly of merit as the means of the relative prediction errors of the gas and water flowrates were improved by 8.9% and 10.9%, respectively.

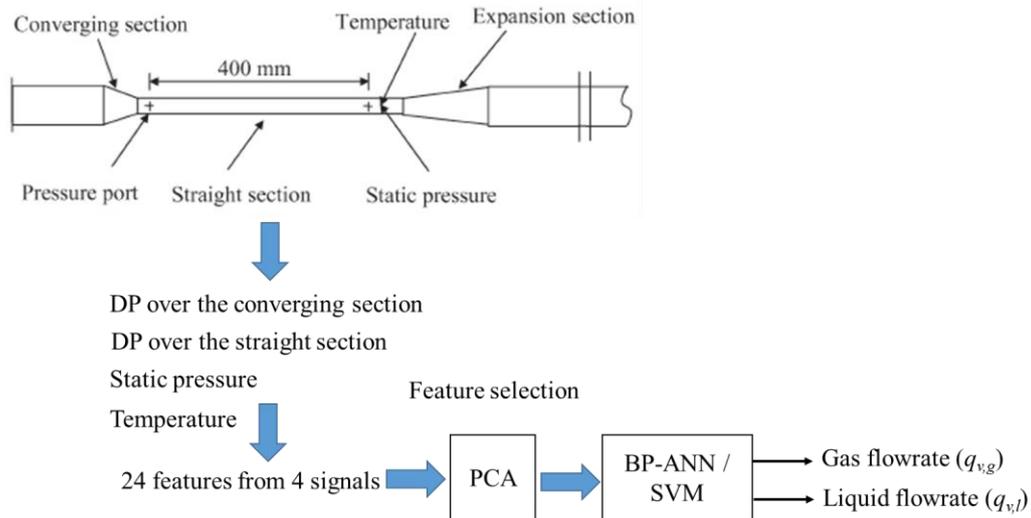


Fig.9 Measurement system based on a throat-extended Venturi meter with ANN/SVM [40]

Shaban and Tavoularis [41] proposed a method for the measurement of gas and liquid flow rates in a vertical upward pipe using differential pressure (DP) signals. As shown in Fig.10, the probability density function and the power spectral density of the normalized DP signals were obtained and processed by principle component analysis (PCA) and independent component analysis (ICA). The two-phase flow regime was firstly identified through the application of the elastic maps method on the probability density function of the DP signals. Then a multi-layer BP-ANN taking the extracted features as inputs was developed for each flow regime (slug, churn and annular) to produce phase flow rates. Experimental tests were conducted with air-water in a vertical pipe of diameter 32.5mm with air superficial velocity between 0.014 m/s and 22 m/s and liquid velocity from 0.04 m/s to 0.4 m/s. Experimental results suggested that the average relative errors of liquid flowrates for slug, churn, annular flow regimes were -0.3%, -0.1% and -0.4%, respectively, and the average relative errors of gas flowrate were 5.5%, 0.5% and 0.6%, respectively.

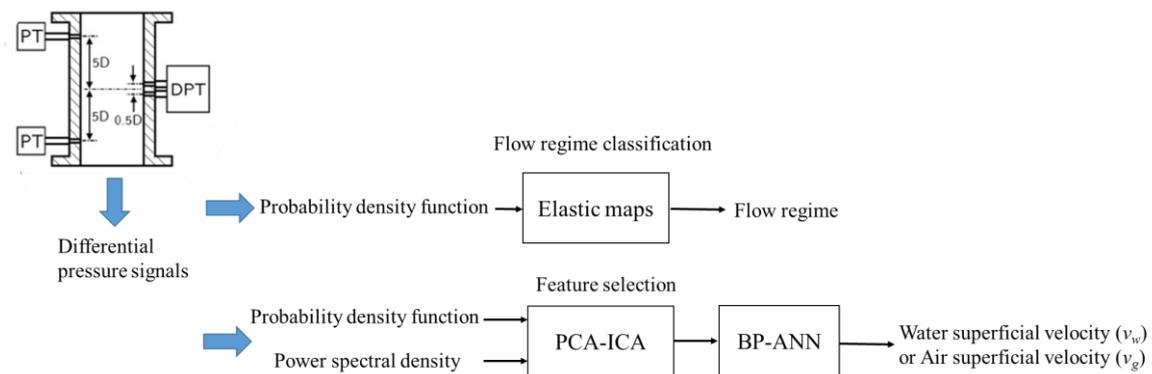


Fig.10 Measurement system based on a differential pressure transducer with elastic maps, PCA-ICA and ANN [41]

### 3.3 Electrical Sensors

Fan and Yan [42] presented an ANN based method to obtain gas and liquid flowrates of two-phase air-water slug flow in a 50 mm bore horizontal pipe through conductance probes. It can be seen from Fig.11 that five characteristic parameters of the mechanistic slug flow model, including translational velocity, slug holdup, film holdup, slug length and film length were extracted from the conductance signals and taken as inputs to the neural network. A feed forward neural network with ten hidden nodes was adopted to predict gas and liquid flowrates. Experimental assessment of the measurement system was conducted with air superficial velocity from 0.58 to 1.86 m/s and water superficial velocity between 0.35 to 1.62 m/s. Results suggested that the overall performance was within  $\pm 10\%$  of full scale for the prediction of both liquid and gas flow rates.

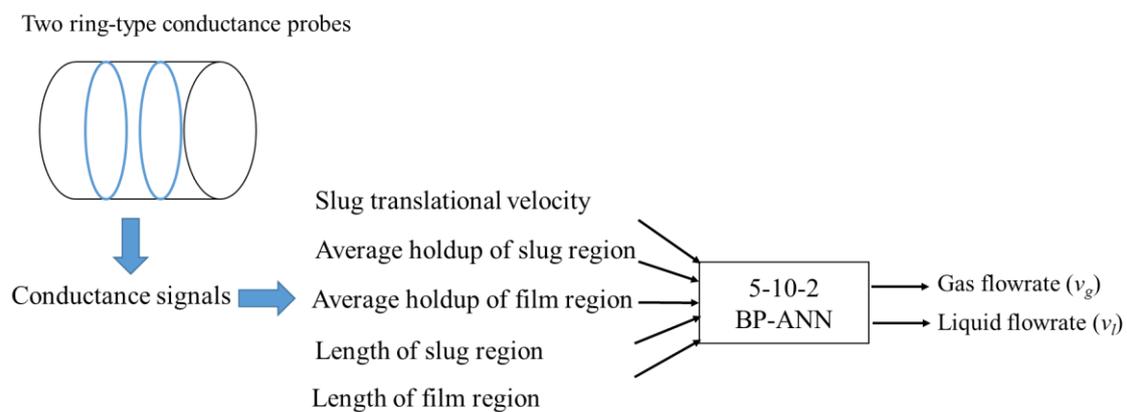


Fig.11 Measurement system based on two ring-type conductance probes and ANN

Yan et al. [43] proposed a novel approach to measure the velocity and mass flowrate of pneumatically conveyed solids using a single ring-shaped electrostatic sensor and BP-ANNs. As shown in Fig. 12, a total of nine feature parameters of the electrostatic signals were extracted in the time and frequency domains. Through feature selection with PCA, two three-layer BP ANNs was developed to estimate the solid velocity and mass flowrate, respectively. Experimental tests were conducted with salt particles on a 50 mm bore test rig. The expected velocity was ranging from 10 to 30 m/s. Results demonstrated that the relative errors for both particle velocity and mass flowrate measurements were mostly within  $\pm 15\%$ . The standard deviations of the relative errors for both measurements are 7.7% and 6.8%, respectively.

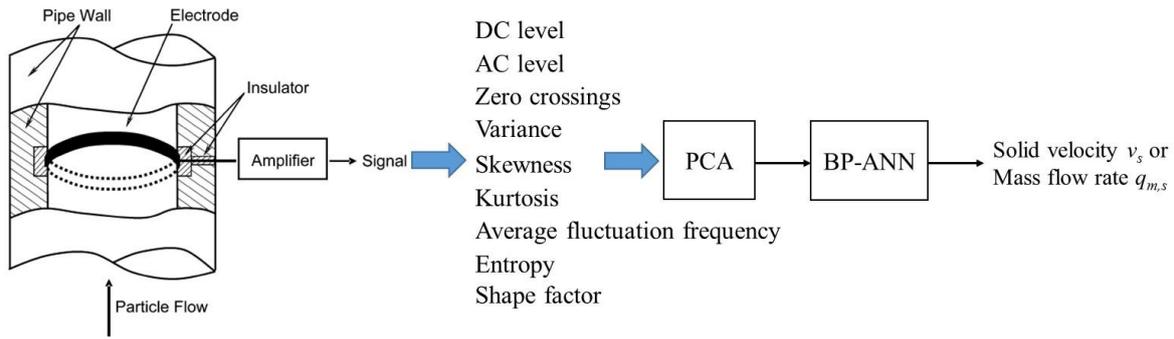


Fig. 12 Measurement system based on an electrostatic sensor and ANN [43]

### 3.4 Optical Sensors

Li et al. [44] applied a laser source, a 12×6 photodiode array sensor and an SVM model to quantify the void fraction of gas-liquid two-phase flow in small channels. As shown in Fig.13, the extracted features from the mean value and standard deviation of the 72 measured signals were taken as inputs of the SVM model. Through experimental tests with Nitrogen-water flow on a horizontal pipe with inner diameters of 4.22, 3.03, 2.16 and 1.08 mm, the flow patterns, including bubble flow, slug flow, stratified flow and annular flow, were observed. The maximum absolute error of the void fraction was found to be 7%.

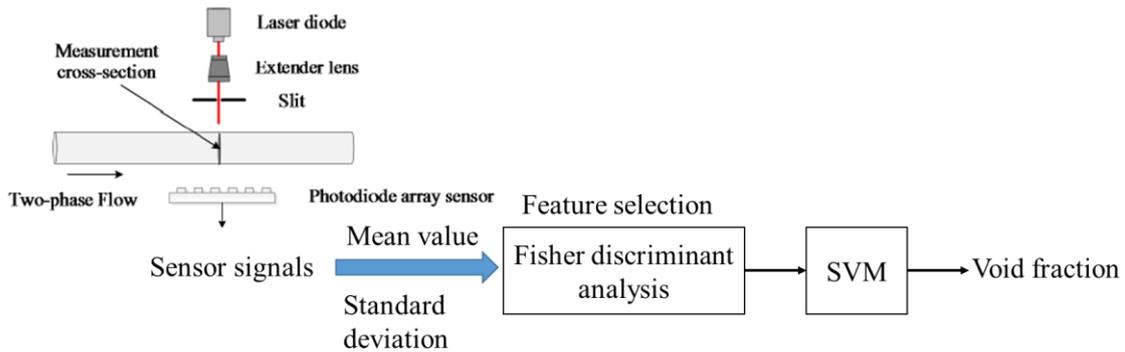


Fig.13 Measurement system based on a laser source, photodiode array sensor and SVM [44]

### 3.5 Combination of Multiple Sensors

Zheng et al. [45] proposed a measurement system using a turbine flowmeter, conductance sensors and SVM to identify the flow pattern and obtain the water cut of air-water two-phase flow in a vertical upward pipe with an inner diameter of 18mm. As shown in Fig.14, the flow pattern was identified through chaotic attractor morphologic analysis of the conductance signals. The total flowrate of the mixture was obtained from the rotating speed of the turbine through polynomial regression. To estimate the water cut of the mixed flow, a total of 10 features, extracted from fluctuant conductance signals in both time and frequency domains

together with the average of rotating speed of the turbine were taken as inputs of the SVM model. The total flowrate of gas-water flow ranged from 0.1 to 2.5 m<sup>3</sup>/h during the tests. Experimental results suggested that the success rate of the flow pattern identification was higher than 96% and the average errors of the water flowrate and gas flowrate measurements were 7.36%.

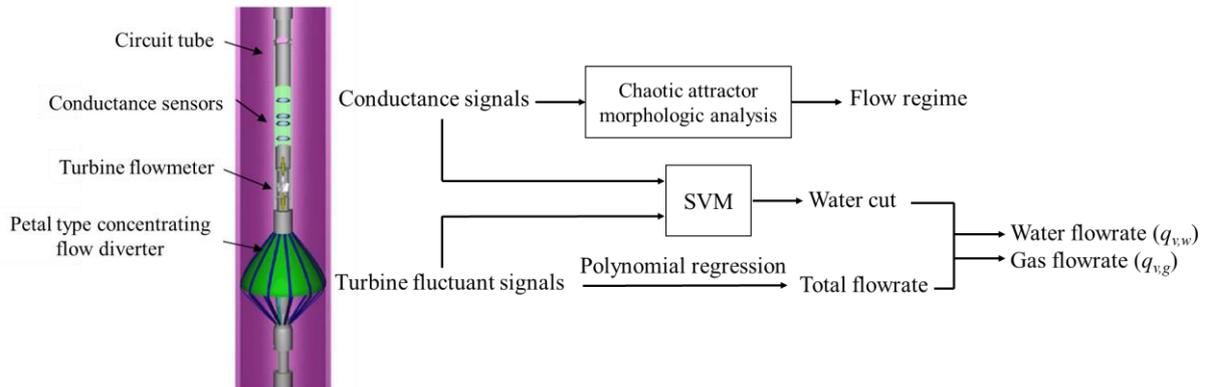
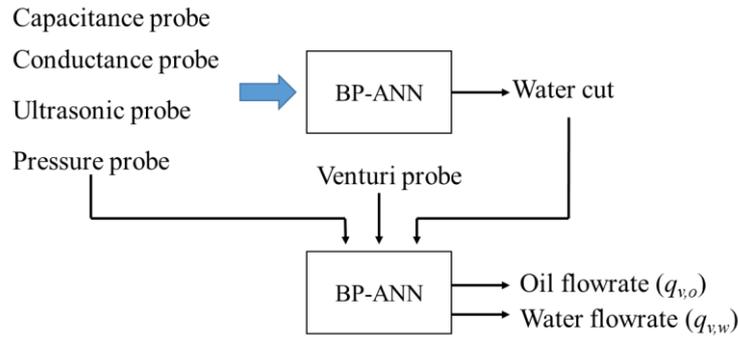
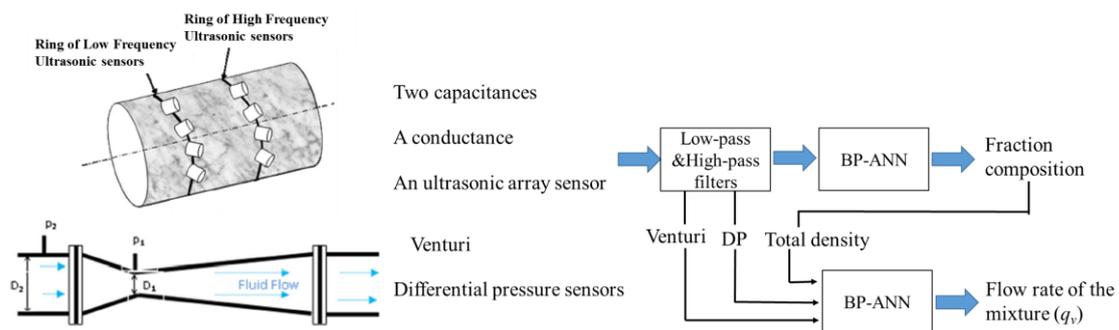


Fig.14 Measurement system based on conductance sensors, a turbine flowmeter and SVM [45]

Meribout et al. [46] integrated impedance measurements with ultrasonic measurements to provide the volumetric flowrate of oil-water flow with the water-cut ranging from 0 to 100%. As shown in Fig.15 (a), a pattern recognition algorithm based on an ANN was implemented. In the first stage, a feedforward three-layer ANN was developed with the signals extracted from the conductance, capacitance, ultrasonic and Venturi probes and an output of the mixture density. With the usage of the ANN, the water cut can be interpreted by the sensor signals. In the second stage, another three-layer ANN was established to obtain oil and water flowrates by combining the estimated water-cut from the first stage and the differential pressure and Venturi outputs. Experimental work was conducted on a two-inch oil-water test rig and the results showed that, for both water-cut and total flow rate determinations, the relative error was less than 5% for any flow regimes. In subsequent research, they applied the same measurement method to oil-gas-water three-phase flow [47]. As shown in Fig.15 (b), two rings of high and low frequency ultrasonic sensors were used for low and high gas fractions, respectively. In this case, an ANN was developed in the first stage with the signals from capacitance, conductance, ultrasound, pressure and Venturi sensors. Then the flow rate of the mixture was obtained using the estimated total density, differential pressure and Venturi outputs. The experimental results demonstrated that the average relative errors were 3.91% for water flowrate, 4.68% for gas flowrate and 6.2% for oil flowrate.



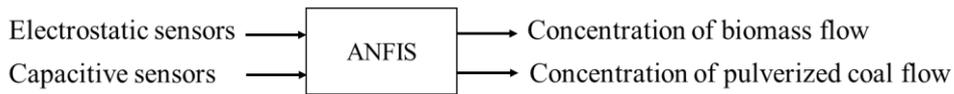
(a) Measurement system for oil-water two-phase flow measurement



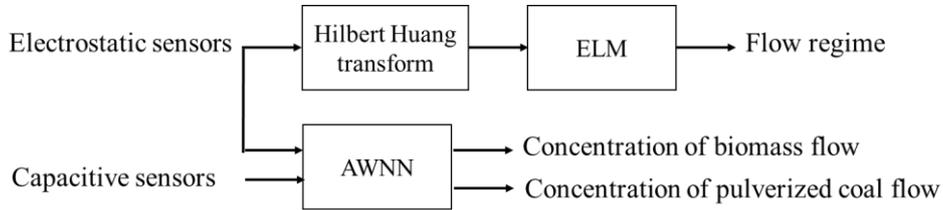
(b) Measurement system for oil-air-water three-phase flow measurement

Fig.15 Measurement system based on capacitance, conductance, ultrasonic, DP sensors and a Venturi flowmeter [46,47]

Wang et al. [48] proposed a data fusion method by combining capacitive and electrostatic sensors to realize online volumetric concentration measurement of pulverized coal/biomass fuel flow in co-fired power plants (Fig.16). An adaptive network based fuzzy inference system (ANFIS) was developed through training with gradient descent method and hybrid method by combining the Kalman filter algorithm with a gradient descent algorithm. Experimental results on a 36 mm bore horizontal quartz glass pipe showed that the ANFIS based on the hybrid learning rule outperformed the system based on the gradient descent learning rule and the fiducial errors of biomass and pulverized coal flows were 1.2% and 0.7%, respectively. Following this research, an extreme learning machine (ELM) based on the electrostatic fluctuation signals was applied to identify the flow regime of coal/biomass/air three-phase flow and an adaptive wavelet neural network (AWNN) based on electrostatic and capacitance sensors was created to predict the volume concentration of each phase [49]. Experimental work was carried out on a 94 mm bore horizontal quartz glass pipe. This method yielded 2.1% fiducial error for biomass and 1.2% for pulverized coal. It was claimed that the training time, identification time and prediction time were much less than other methods.



(a) Measurement system based on ANFIS



(b) Measurement system based on ELM and AWNN

Fig.16 Measurement system based on capacitive and electrostatic sensors

### 3.6 Coriolis Flowmeters

Liu et al. [50] used a neural network to estimate the mass flow error of a 1-inch Coriolis mass flowmeter on a horizontal pipe. As shown in Fig.17, the multi-layer perceptron and radial basis function networks accept four inputs, including temperature, damping, density drop and flowrate to estimate the mass flow error. Then the mass flow readings from the Coriolis flowmeter are corrected with the estimated mass flow error. Experimental tests were conducted with the liquid mass flowrate ranging from 1.5 to 3.6 kg/s and density drop up to 35%. Although most of the mass flow errors were reduced to within  $\pm 2\%$ , the gas entrainment was not quantified and other installation conditions were not considered.

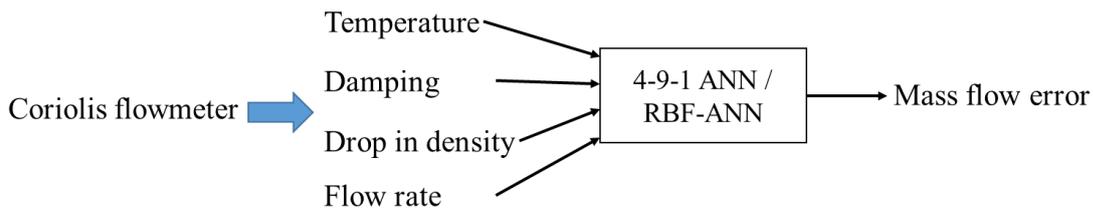


Fig.17 Measurement system based on Coriolis flowmeter and ANN

Using a Coriolis mass flowmeter and a similar correction model, Mattar et al. [51] and Henry et al. [52] reported a case study of two-phase flow metering of heavy oil. Trials were carried out on a 75 mm bore flowmeter with a mass flowrate between 1 kg/s and 10 kg/s and gas void fraction up to 80%. Experimental results demonstrated that the corrected measurements were typically within 1-5% of the nominal mass flow and density for both steady and slugging two-phase flows.

Subsequently, the field tests using Coriolis flowmeters incorporating the ANN correction model were conducted to measure slugging two-phase CO<sub>2</sub> flow [53]. Due to the pressure losses through the pipeline network, the well-heads received the CO<sub>2</sub> at the pressure of 5.52-7.03MPa and the temperature of 4-32°C. The significant variations in pressure and temperature resulted in gas-liquid two-phase CO<sub>2</sub> flow. In this case, the comparison results showed a 5% difference between the Coriolis flowmeters and the Pecos station sales meter.

Apart from two-phase flow tests, Henry et al. [54] described another empirical method by combining a Coriolis mass flowmeter with a commercial water cut meter (Weatherford Red eye MP water cut meter) to achieve three-phase flow measurement. The water cut meter was used to indicate the proportion of water in the liquid flow. Experimental tests were undertaken on a 50 mm bore test rig with the total liquid flowrate ranging from 2.4 kg/s up to 11 kg/s, the water cut from 0% to 100% and gas volume fraction between 0% and 50%. Results demonstrated that the total liquid mass flow error was reduced within  $\pm 2.5\%$  and the gas mass flow error within  $\pm 5\%$ . The oil mass flow error limit was  $\pm 6\%$  for water cut less than 70% and  $\pm 15\%$ , for water cut between 70% and 95%.

A method based on fuzzy inference was proposed to correct the mass flow error of a Coriolis mass flowmeter for two-phase flow measurement [55]. The system accepted damping, density drop and apparent mass flowrate as inputs to the fuzzy inference system to generate the corrected mass flowrate. Lari and Shabaninia [56] applied a neuro-fuzzy algorithm to correct the error of a Coriolis mass flowmeter for air-water two-phase flow measurement. However, the experimental data and results were not explained in detail in their reports.

In order to maintain the flow tube oscillation under two-phase flow conditions, Hou et al. [57] developed a digital Coriolis flow transmitter and tested on a commercial 1-inch Coriolis flowmeter. The measurement errors under gas-liquid two-phase flow conditions were corrected using a feed-forward neural network with two inputs - apparent liquid mass flowrate and observed density drop. The online correction results showed that, when water flowrate varied from 3 to 15 kg/min with gas volume fraction up to 25%, the flowrate errors were within  $\pm 3.5\%$  while density errors were within  $\pm 1.5\%$ .

A 25 mm bore Coriolis mass flowmeter together with SVM algorithms was applied by Ma et al. [58] to measure the overall mass flowrate of oil-water two-phase flow. The oil flowrate ranged from 0.27 to 5 m<sup>3</sup>/h and water flowrate between 0.2 and 7 m<sup>3</sup>/h. Experimental results showed that the relative error of the total mass flowrate was within  $\pm 1\%$  while the individual mass flowrate had the maximum error of  $\pm 8\%$ .

Further investigations into the application of soft computing techniques to multiphase measurement were conducted by Wang et al. [59]. A 25 mm bore Coriolis mass flowmeter was

tested with air-water flow on horizontal and vertical pipe sections. Different data-driven models based on ANN, SVM and GP were developed, respectively, to estimate the liquid mass flowrate and gas volume fraction. The inputs of the data-driven models are determined through input variable selection methods [60]. For the estimation of liquid mass flowrate, the input variables take apparent mass flowrate, observed density, damping and DP, while for the gas volume fraction (GVF) prediction, the apparent mass flowrate, density and DP are taken as inputs. Experimental tests were conducted with liquid mass flowrate from 700 kg/h to 14500 kg/h and gas volume fraction between 0 and 30%. The performance of the data-driven models was assessed with air-water two-phase flow and the results demonstrated that SVM outperformed ANN and GP. For liquid mass flowrate measurement with the SVM models, 93.49% of the experimental data yield a relative error less than  $\pm 1\%$  on the horizontal pipeline, while 96.17% of the results are within  $\pm 1\%$  on the vertical installation. The SVM models predict the gas volume fraction with a relative error less than 10% for 93.10% and 94.25% of the test conditions on the horizontal and vertical installations, respectively.

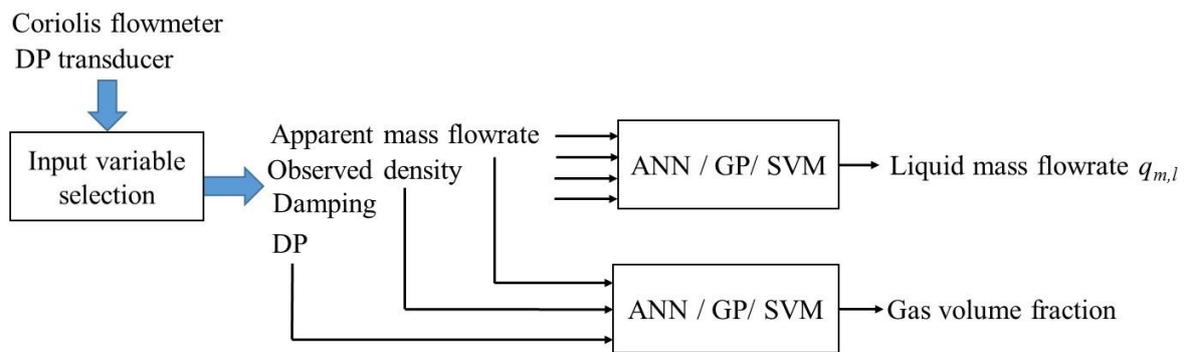


Fig. 18 Measurement system based on Coriolis flowmeter and ANN/GP/SVM

Afterwards, Wang et al. [61] applied a 15mm bore Coriolis mass flowmeter together with a BP-ANN model to the measurement of gas-liquid two-phase carbon dioxide flow. The ANN accepts two inputs including apparent mass flowrate and observed density drop and output total CO<sub>2</sub> mass flowrate. Experimental evaluation was conducted with the liquid mass flowrate between 300 kg/h and 3050 kg/h and gas mass flowrate from 0 to 330 kg/h. Experimental results have suggested that the Coriolis flowmeter with the developed correction method is capable of providing the mass flowrate of gas-liquid CO<sub>2</sub> flow with errors mostly within  $\pm 2\%$  and  $\pm 1.5\%$  on horizontal and vertical pipelines, respectively.

### 3.7 Summary

This section has reviewed the indirect multiphase measurement systems incorporating traditional sensors and soft computing techniques for the measurement of phase flowrates and phase fractions. The main characteristic of soft computing techniques is to provide solutions to real world problems which cannot be modelled or are too difficult to model mathematically. Multiphase flow is very complex and difficult to understand and model due to the simultaneous presence and interactions between different phases or components in the same stream. For this reason, a range of soft computing techniques such as ANN, SVM, ANFIS and GP have been

applied to estimate the flow rates or phase fractions. Experimental studies have been conducted to evaluate the validity of soft computing techniques for the measurement of gas-liquid, gas-solid two-phase or three-phase flows.

## **4. Trends and Future Developments**

### **4.1 Sensor Fusion**

Traditional sensors incorporating soft computing techniques provide an effective solution to the measurement of phase flowrates and phase fractions. Table 2 summarizes the sensors used, soft computing methods, experimental conditions and measurement errors of the indirect measurement systems discussed in this review. It should be noted that ‘GVF variation’ in Table 2 represents the absolute error of the measurement from the reference value while the rest of the results is the averaged relative error in the measurements from the reference values. Among these measurement systems, it is obvious that multi-sensors are capable of estimating more parameters with higher accuracies. For example, conductance sensors incorporating an ANN can measure the air-water flow with an error less than  $\pm 10\%$  while conductance sensors together with a Turbine flowmeter are able to obtain the velocity of individual phase with an error no greater than 7.36%. A combination of conductance, capacitance, ultrasonic, DP sensors and a Venturi meter may be applied to measure oil-air-water three-phase flow. Sensor fusion is an effective solution to extending the measurement range and applicability of conventional flow instruments. In addition, it is clear that Coriolis mass flowmeters provide more accurate mass flowrate measurement in comparison with other flowmeters. Coriolis mass flowmeters in combination with sensor fusion are expected to have much potential for future multiphase flow measurement [62].

Table 2 Indirect measurement approaches using traditional sensors and soft computing techniques

Sensors	Soft computing method	Multiphase flow	Pipe diameter (mm)	Average error
Ultrasonic sensor [39]	ANN	air-oil	25, 50	GVF variation: $\pm 4.2$
Throat-extended Venturi [40]	ANN	natural gas-water	50	$q_{v,g}$ : 3.14%, $q_{v,l}$ : 4.77%
Throat-extended Venturi [40]	SVM	natural gas-water	50	$q_{v,g}$ : 2.86%, $q_{v,l}$ : 4.25%
DP sensors [41]	ANN	air-water	32.5	$v_g$ : 2.4%, $v_l$ : -0.3%
Conductance sensors [42]	ANN	air-water	50	$v_g, v_l$ : $\leq \pm 10\%$
Laser diode, photodiode array [44]	SVM	nitrogen-water	4.22, 3.03, 2.16, 1.08	GVF variation $< 7\%$
Conductance+Turbine flowmeter [45]	ANN	air-water	18	$v_g, v_l$ : 7.36%
Capacitance+Conductance+Ultrasonic+DP+Venturi[46]	ANN	oil-water	50	$v$ : $< 5\%$
Capacitance+Conductance+Ultrasonic+DP+Venturi[47]	ANN	oil-air-water	50	$v_o$ : 6.2%, $v_g$ : 4.68%, $v_l$ : 3.91%
Coriolis flowmeter [50]	ANN	air-water	25	$q_{m,l}$ : $\leq \pm 2\%$
Coriolis flowmeter [52]	ANN	air-heavy oil	75	$q_{m,l}$ : $< 1\% \sim 5\%$
Coriolis flowmeter [53]	ANN	gas-liquid CO <sub>2</sub>	-	$q_{m,l}$ : 5%
Coriolis flowmeter+water cut meter [54]	ANN	oil-air-water	50	$q_{m,l}$ : $\pm 2.5\%$ , $q_{m,g}$ : $\pm 5\%$ , $q_{m,o}$ : $\pm 6\%$ , $\pm 15\%$
Coriolis flowmeter [55]	Fuzzy system	air-water	-	-
Coriolis flowmeter [56]	Neuro-Fuzzy	air-water	-	-
Coriolis flowmeter [57]	ANN	air-water	25	$q_{m,l}$ : $\leq \pm 3.5\%$
Coriolis flowmeter [58]	SVM	oil-water	25	$q_m$ : $\leq \pm 1\%$ , $q_{m,o}, q_{m,w}$ : $\leq \pm 8\%$
Coriolis flowmeter+DP [59]	ANN / SVM /GP	air-water	25	$q_{m,l}$ : $\leq \pm 2\%$ , GVF: $\leq \pm 10\%$
Coriolis flowmeter [61]	ANN	gas-liquid CO <sub>2</sub>		$q_{m,l}$ : $\leq \pm 2\%$
Electrostatic sensor [43]	ANN	salt-air	50	$v_s, q_{m,s}$ : $\leq \pm 15\%$
Capacitance+Electrostatic sensor [48]	ANFIS	coal-biomass-air	36	$C_b$ : 1.2%, $C_c$ : 0.7%
Capacitance+Electrostatic sensor [49]	ELM	coal-biomass-air	94	$C_b$ : 2.1%, $C_c$ : 1.2%

## 4.2 Soft Computing Techniques

MLP neural networks have been widely used for the estimation of individual phase flowrate and phase fraction. However, the structure parameters of neural networks are required to be adjusted during the training process and normally determined through trial-and-error. Due to the fixed structure and less adjustable parameters, RBF neural networks were employed in some research to improve high training efficiency. Although neural networks have provided effective solutions to multiphase flow measurement, ANN is based on empirical risk minimization and all the parameters are tuned iteratively, so that ANN may suffer from overfitting. In this case, SVM based on structural risk minimization offers an alternative option. Some research work [40, 59] has proven that SVM achieved better performance than ANN in terms of generalization ability. As for evolutionary algorithms, GA has been widely used to optimize the internal parameters of ANN. In comparison with ANN and SVM, there are less applications of fuzzy logic and probabilistic reasoning to multiphase flow measurement. In addition, some knowledge based systems combining ANN and fuzzy logic such as ANFIS have been developed [56,48]. This kind of hybrid system takes the advantages of ANN and fuzzy logic systems. Soft computing approaches are preferable to the conventional methods for solving problems which are difficult to describe by analytical or mathematical models. The successful applications suggest that soft computing techniques will have an increasingly greater impact on multiphase flow metering in the coming years. Hybrid models by exploiting the strength and advantages of each technique offer a new dimension in the field of multiphase flow measurement. In addition, deep learning is a set of machine learning algorithms that model high-level abstractions in data using architectures consisting of multiple nonlinear transformations [63,64]. Deep learning has been successfully applied in the fields of computer vision, speech recognition and social network filtering [64]. Using deep learning for flow pattern recognition has recently been attempted [65,66]. It is expected that more applications of deep learning to multiphase flow measurement would emerge over the next few years.

From the description in Section 3, it is clear that a number of sensing components incorporating soft computing techniques have been applied for the measurement of individual flowrate and phase fraction in gas-liquid and liquid-liquid flows. Some research focuses on the measurement of individual solid phase concentration in gas-solid three-phase flow. However, the individual flowrate has not been quantified. There are few reports on the measurement of phase flowrates or phase fractions of slurry flow using soft computing techniques. However, the parameters for considerations of pipe design, such as pressure drop [67], hold-up [68,69] and critical velocity [70,71] have been predicted with soft computing techniques. It is anticipated that more progress in developing the measurement systems incorporating traditional sensors and soft computing techniques would be achieved for the measurement of gas-liquid, gas-solid and liquid-solid flows in the next few years.

### 4.3 Data-Driven Modelling

An empirical model developed with soft computing methods is normally regarded as a data-driven model. Data-driven models would be useful in solving a practical problem or modelling a particular system or process if a considerable volume of data describing this problem is available, provided that there are no significant changes to the modelled system during the period covered by the model [72]. However, the primary challenge in real-world applications of the data-driven models lies in that the models established from experimental data under limited laboratory conditions may not be applicable to the process conditions where the flow metering systems are installed. Factors that may affect the performance of the models under real-world conditions include pipe diameter, pipeline orientation, ranges of flow rates and phase fractions, temperature, pressure and viscosity of the fluids being measured. Certain in-field training of the data-driven models may be undertaken, provided that there are reliable reference data or history data are available.

In order to develop an optimal model for the desired output and enhance the generalization ability of the model, two aspects should be taken into consideration, namely, input variable selection and model evaluation. Input variable selection is to extract useful information from the available data and identify suitable input variables which are able to well explain the desired output for a data-driven model. Through input variable selection to eliminate the irrelevant or redundant variables, the complexity of the models structure is simplified and the computational efficiency is improved. The models [40,41,43,44,60] include feature selection or variable selection and hence yield better performance. Hereby, input variable selection is an important step in data-driven modelling. The comprehensive review of input variable selection methods for artificial neural networks by May et al. [73] offers useful guidance.

Data-driven modelling is a process to find connections between the input variables and the desired outputs through analysing the available data, even without any explicit knowledge of the physical behaviour of the system. Once the model is trained, it can be tested using an independent data set to determine how well it can generalise to unseen data. It can be seen from Xu et al. [40] and Wang et al. [59] that different models with the same experimental data perform differently. Even though the model structure is determined, different model parameters also affect the performance of the data-driven models. In the case of an ANN the parameters such as weights, biases and transfer functions have significant influences on the outputs of the ANN. Similarly, the performance of an SVM model depends on the penalty parameters and kernel functions used. For this reason, several optimization algorithms such as GA [74], response surface method [75] and central composite design [76], are used to obtain the optimal parameters for data-driven models. In order to achieve a better solution to the problem, model evaluation is also an important process. Therefore, it is suggested that input variable selection methods and model evaluation should be carefully considered in the developing process of data-driven models for multiphase flow measurement or other applications.

#### **4.4 Product Enhancement and New Applications**

Soft computing techniques extend the usability of traditional sensors from single phase to multiphase flow measurement. Such techniques are superior to conventional methods to solve challenging problems which are difficult to describe by analytical or mathematical models. By incorporating soft computing techniques, ultrasonic sensors, Venturi flowmeters, conductance sensors and Coriolis flowmeters are capable of measuring the individual flowrates or phase fractions in two-phase or three-phase flow. The techniques provide an efficient, intelligent and cost-effective approach to developing multiphase flowmeters. It is thus expected that the measurement systems combining conventional sensors and soft computing techniques will be applied to multiphase flow metering in the oil and gas sector, CCS systems, coal/biomass fired power plant, slurry processes and bunkering centres.

It should be pointed out that calibration of multiphase flowmeters incorporating soft computing models is a changing issue. In order to achieve the same performance of the flowmeters in practical applications as in the training process, this method requires primary calibration on a flow test facility and in-situ adjustment of the outputs during field operations.

#### **5. Conclusions**

This review has attempted to present the applications of soft computing techniques to multiphase flow measurement and define the state-of-the-art in the development of multiphase flowmeters incorporating such techniques. This review covers the research which have been conducted within the past 15 years and focuses on the measurement of phase flowrates and phase fractions using conventional sensors incorporating soft computing techniques.

Multiphase flow measurement is a complex and difficult problem for engineers, practitioners and academics. Although extensive research has been conducted on sensor design, signal processing and development of test facilities, it is still a long way to develop an ideal multiphase flowmeter. With advances of soft computing techniques in handling imprecise, uncertain, ambiguous, incomplete and subjective data and information, data-driven models and hybrid models based on soft computing techniques provide effective solutions to the measurement problems. It is envisioned that soft computing techniques will have a more important part to play in multiphase flow measurement.

#### **Acknowledgement**

The authors would like to acknowledge the National Natural Science Foundation of China (No. 61573140), the Fundamental Research Funds for the Central Universities (No. 2016MS23) and China Postdoctoral Science Foundation (No. 2015M581045) for providing financial support for this research.

## References

- [1] R. Baker, Flow measurement handbook-industrial designs, operating principles, performance, and applications, second ed., Cambridge University Press, 2016.
- [2] M. Rahimi-Gorji, T. B. Gorji, M. Gorji-Bandpy, Details of regional particle deposition and airflow structures in a realistic model of human tracheobronchial airways: two-phase flow simulation, *Comput. Biol. Med.* 74 (2016) 1-17.
- [3] M. Rahimi-Gorji, O. Pourmehran, M. Gorji-Bandpy, T. B. Gorji, CFD simulation of airflow behaviour and particle transport and deposition in different breathing conditions through the realistic model of human airways, *J. Mol. Liq.* 209 (2015) 121-133.
- [4] R. Thorn, G. Johansen, E. Hammer, Recent developments in three-phase flow measurement, *Meas. Sci. Technol.* 8 (1997) 691-701.
- [5] R. Thorn, G. Johansen, B. Hjertaker, Three-phase flow measurement in the petroleum industry, *Meas. Sci. Technol.* 24 (2013) 1-17.
- [6] G. Falcone, G. Hewitt, C. Alimonti, B. Harrison, Multiphase flow metering: current trends and future developments, *J. Petrol. Technol.* 54 (2002) 77-84.
- [7] Y. Yan, Mass flow measurement of bulk solids in pneumatic pipelines, *Meas. Sci. Technol.* 7 (1996) 1687-1706.
- [8] Y. Zheng, Q. Liu, Review of certain key issues in indirect measurements of the mass flow rate of solids in pneumatic conveying pipelines, *Meas.* 43 (2010) 727-734.
- [9] J. Sun, Y. Yan, Non-intrusive measurement and hydrodynamics characterization of gas-solid fluidized beds: a review, *Meas. Sci. Technol.* 27 (2016) 1-31.
- [10] K. Albion, I. Briens, C. Briens, F. Berruti, Multiphase flow measurement techniques for slurry transport, *Int. J. Chem. React. Eng.* 9 (2011) 18-19.
- [11] L. Zadeh, Soft computing and fuzzy logic, *IEEE Software.* 11 (1994), 48-56.
- [12] P. Zarchan, H. Musoff, Fundamentals of Kalman filtering: a practical approach, fourth ed., American Institute of Aeronautics and Astronautics, 2015.
- [13] S. Das, A. Kumar, B. Das, A. Burnwal, On soft computing techniques in various areas, in: National Conference on Advancement of Computing in Engineering Research, 2013.
- [14] M. Meireles, P. Almeida, M. Simoes, A comprehensive review for industrial applicability of artificial neural networks, *IEEE Trans. Ind. Electron.* 50 (2003) 585-601.
- [15] M. Hagan, H. Demuth, M. Beale, O. Jesus, Neural Network Design, second ed., 2011.
- [16] Kevin. Gurney, An Introduction to Neural Networks, Taylor & Francis e-Library, 2004.

- [17] G.J. Bowden, H.R. Maier, G.C. Dandy, Input determination for neural network models in water resources applications. Part 2. Case study: Forecasting salinity in a river, *J. Hydrol.* 301 (2005) 93-107.
- [18] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning* 20 (1995) 273-297.
- [19] H. Drucker, C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, *Neural Information Processing System* 9 (1997) 155-161.
- [20] M. Jordan, J. Kleinberg, B. Scholkopf, *Support Vector Machines*, Information Science and Statistics, New York: Springer, 2008.
- [21] A. Ben-Hur, D. Horn, H. T. Siegelmann, V. Vapnik, Support vector clustering, *J. Mach. Learn. Res.* 2 (2001) 125-137.
- [22] J. Suykens, J. Vandewalle, Least squares support vector machine classifiers, *Neural Process. Lett.* 9 (1999) 293-300.
- [23] R. Jayadeva, S. Khemchandani, Chandra, Twin support vector machines for pattern classification, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (2017) 905-910.
- [24] M. Gonen, E. Alpaydin, Multiple kernel learning Algorithms, *J. Mach. Learn. Res.* 12 (2011) 2211-2268.
- [25] C. Lin, S. Wang, Fuzzy support vector machines, *IEEE Trans. Neur. Net.* 13 (2002) 464-471.
- [26] Y. Tian, Y. Shi, X. Liu, Recent advances on support vector machines research, *Technol. Econ. Dev. Eco.* 18 (2012) 5-33.
- [27] A. Abraham, N. Nedjah, L. Mourelle, *Genetic Systems Programming Theory and Experiences*, 2006.
- [28] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Evolution*, Cambridge, MA, USA: MIT Press, 1992.
- [29] J. Madár, J. Abonyi, F. Szeifert, Genetic Programming for the identification of nonlinear input-output models, *Ind. Eng. Chem. Res.* 44 (2005) 3178-3186.
- [30] L. Zadeh, Fuzzy sets, *Information and Control* 8 (1965) 338-353.
- [31] O. Castillo, P. Melin, A review on interval type-2 fuzzy logic applications in intelligent control, *Information Science*, 279 (2014) 615-631.
- [32] T. Koski, J. Noble, A review of Bayesian networks and structure learning, *Mathematic Application* 40 (2012) 51-103.
- [33] S. Park, L. Park, C. Park, A neuro-genetic controller for nonminimum phase systems, *IEEE Trans. Neur. Net.* 6 (1995) 1297-1300.

- [34] F. Chang, Y. Chang, Adaptive neuro-fuzzy inference system for prediction of water level in reservoir, *Adv. Water Resour.* 29 (2006) 1-10.
- [35] J. Buckley, Fuzzy genetic algorithm and applications, *Fuzzy Set. Syst.* 61 (1994) 129-136.
- [36] P. Melin, O. Castillo, Intelligent control of complex electrochemical systems with a neuro-fuzzy-genetic approach, *IEEE Trans. Ind. Electro.* 48 (2001) 951-955.
- [37] J. Jang, ANFIS: Adaptive-Network-Based fuzzy inference system, *IEEE Trans. Syst. Man Cyb.* 23 (1993) 665-684.
- [38] A. Depari, A. Flammini, D. Marioli, A. Taroni, Application of an ANFIS algorithm to sensor data processing, *IEEE Trans. Instrum. Meas.* 56 (2007) 75-79.
- [39] M. Figueiredo, J. Goncalves, A. Nakashima, A. Fileti, R. Carvalho, The use of an ultrasonic technique and neural networks for identification of the flow pattern and measurement of the gas volume fraction in multiphase flows, *Exp. Therm. Fluid Sci.* 70 (2016) 29-50.
- [40] L. Xu, W. Zhou, X. Li, S. Tang, Wet gas metering using a revised Venturi meter and soft-computing approximation techniques, *IEEE Trans. Instrum. Meas.* 60 (2011) 947-956.
- [41] H. Shaban, S. Tavoularis, Measurement of gas and liquid flow rates in two-phase pipe flows by the application of machine learning techniques to differential pressure signals, *Int. J. Multiph. Flow* 67 (2014) 106-117.
- [42] S. Fan, T. Yan, Two-phase air-water slug flow measurement in horizontal pipe using conductance probes and neural network, *IEEE Trans. Instrum. Meas.* 63 (2014) 456-466.
- [43] Y. Yan, L. Xu, P. Lee, Mass flow measurement of fine particles in a pneumatic suspension using electrostatic sensing and neural network techniques, *IEEE Trans. Instrum. Meas.* 55 (2006) 2330-2334.
- [44] H. Li, H. Ji, Z. Huang, B. Wang, H. Li, G. Wu, A new void fraction measurement method for gas-liquid two-phase flow in small channels, *Sensors* 16 (2016) 1-13.
- [45] G. Zheng, N. Jin, X. Jia, P. Lv, X. Liu, Gas-liquid two phase flow measurement method based on combination instrument of turbine flowmeter and conductance sensor, *Int. J. Multiph. Flow* 34 (2008) 1031-1047.
- [46] M. Meribout, N. Al-Rawahi, A. Al-Naamany, A. Al-Bimani, K. Al-Busaidi, A. Meribout, Integration of impedance measurements with acoustic measurements for accurate two phase flow metering in case of high water-cut, *Flow Meas. Instrum.* 21 (2010) 8-19.

- [47] M. Meribout, N. Al-Rawahi, A. Al-Naamany, A. Al-Bimani, K. Al-Busaidi, A. Meribout, A multisensor intelligent device for real-time multiphase flow metering in oil fields, *IEEE Trans. Instrum. Meas.* 59 (2010) 1507-1519.
- [48] X. Wang, H. Hu, A. Zhang, Concentration measurement of three-phase flow based on multi-sensor data fusion using adaptive fuzzy inference system, *Flow Meas. Instrum.* 39 (2014) 1-8.
- [49] X. Wang, H. Hu, X. Liu, Multiphase data fusion techniques with ELM for pulverized-fuel flow concentration measurement in cofired power plant, *IEEE Trans. Instrum. Meas.* 64 (2015) 2769-2780.
- [50] R. Liu, M. Fuent, M. Henry, M. Duta, A neural network to correct mass flow errors caused by two-phase flow in a digital Coriolis mass flowmeter, *Flow Meas. Instrum.* 12 (2001) 53-63.
- [51] W. Mattar, M. Henry, M. Duta, M. Tombs, Multiphase Coriolis flowmeter, Patent US20060161366A1, 2006.
- [52] M. Henry, M. Tombs, M. Duta, F. Zhou, R. Mercado, F. Kenyery, J. Shen, M. Morles, C. Garcia, R. Langansan, Two-phase flow metering of heavy oil using a Coriolis mass flow meter: A case study, *Flow Meas. Instrum.* 17 (2006) 399-413.
- [53] T. Green, M. Reese, M. Henry, Two-phase CO<sub>2</sub> measurement and control in the Yates oil field, *Measurement and Control* 41 (2008) 1-10.
- [54] M. Henry, M. Tombs, M. Zamora, F. Zhou, Coriolis mass flow metering for three-phase flow: a case study, *Flow Meas. Instrum.* 30 (2013) 112-122.
- [55] B. Safarinejadian, M. Tajeddini, L. Mahmoodi, A new fuzzy based method for error correction of Coriolis mass flow meter in presence of two-phase fluid, In: *International Conference on Artificial Intelligence and Image Processing*, 2012.
- [56] V. Lari, F. Shabaninia, Error correction of a Coriolis mass flow meter in two-phase flow measurement using neuro-fuzzy, in: *16th CSI International Symposium on Artificial Intelligence and Signal Processing*, 2012.
- [57] Q. Hou, K. Xu, M. Fang, Y. Shi, B. Tao, R. Jiang, Gas-liquid two-phase flow correction method for digital CMF, *IEEE Trans. Instrum. Meas.* 63 (2014) 2396-2404.
- [58] L. Ma, H. Zhang, H. Zhou, Q. He, Mass flow measurement of oil water two-phase flow based on Coriolis flow meter and SVM, *Journal of Chemical Engineering of Chinese Universities* 21 (2007) 201-205.
- [59] L. Wang, J. Liu, Y. Yan, X. Wang, T. Wang, a. Gas-liquid two-phase flow measurement using Coriolis flowmeters incorporating artificial neural network, support vector

- machine and genetic programming algorithms, *IEEE Trans. Instrum. Meas.* 66 (2017) 852-868.
- [60] L. Wang, Y. Yan, X. Wang, T. Wang, Input variable selection for data-driven models of Coriolis flowmeters for two-phase flow measurement, *Meas. Sci. Technol.* 28 (2017) 1-12.
- [61] L. Wang, Y. Yan, X. Wang, T. Wang, Q. Duan, W. Zhang, Mass flow measurement of gas-liquid two-phase CO<sub>2</sub> in CCS transportation pipelines using Coriolis flowmeters, *Int. J. Greenhouse Gas Control*, 68 (2018) 269-275.
- [62] T. Wang, R. Baker, Coriolis flowmeters: a review of developments over the past 20 years, and an assessment of the state of the art and likely future directions, *Flow Meas. Instrum.* 40 (2014) 99-123.
- [63] Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* 521 (2015) 436-444.
- [64] L. Deng, D. Yu, Deep learning: Methods and Applications. *Foundations and Trends in Signal Processing*, 7 (2014) 1-199.
- [65] I. E. Poletaev, K. S. Pervunin, M. P. Tokarev, Artificial neural network for bubbles pattern recognition on the images. *Journal of Physics: Conference Series*, 754 (2016) 072002.
- [66] M. Ezzatabadipour, P. Singh, M.D. Robinson, P. Guillen-Rondon, C. Torres, Deep learning as a tool to predict flow patterns in two-phase flow, arXiv preprint (2017) arXiv: 1705.07117.
- [67] S. Lahiri, K. Ghanta, Artificial neural network model with the parameter tuning assisted by a differential evolution technique: the study of the hold up of the slurry flow in a pipeline, *Chemical Industry & Chemical Engineering* 15 (2009) 103-117.
- [68] S. Lahiri, K. Ghanta, Development of an artificial neural network correlation for prediction of hold-up of slurry transport in pipelines, *Chemical Engineering Science*, 63 (2008) 1497-1509.
- [69] S. Lahiri, K. Ghanta, Support vector regression with parameter tuning assisted by differential evolution techniques: Study on pressure drop of slurry flow in pipeline, *Korean Journal of Chemical Engineering*, 26 (2009) 1175-1185.
- [70] S. Lahiri, K. Ghanta, Artificial neural network model with parameter tuning assisted by genetic algorithm technique: study of critical velocity of slurry flow in pipeline, *Asia-Pac. J. Chem. Eng.* 5 (2010) 763-777.
- [71] H. Azamathulla, M. ASCE, Z. Ahmad, Estimation of critical velocity for slurry transport through pipeline using adaptive neuro-fuzzy interference system and gene-expression programming, *J. Pipeline Syst. Eng.* 4 (2013) 131-137.

- [72] D. Solomatine, L.M. See, R.J. Abrahart, Data-driven modelling: concepts, approaches and experiences. Chap. 2 In Practical Hydroinformatics. Springer Berlin Heidelberg: Water Science and Technology Library, 68 (2008) 17-30.
- [73] R. May, G. Dandy, H. Maier, Review of input variable selection methods for artificial neural networks. Artificial Neural Networks-Methodological Advances and Biomedical Applications (InTech) Chapter 2, 2011.
- [74] E. Turunen, K. Raivio, T. Mantere, Soft computing methods. Mathematical Modelling, (2016) 79-112.
- [75] M. Rahimi-Gorji, M. Ghajar, A.H. Kakaee, D. D. Ganji, Modeling of the air conditions effects on the power and fuel consumption of the SI engine using neural networks and regression. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 39 (2017) 375-384.
- [76] M. Rahimi-Gorji, O. Pourmehran, M. Hatami, D.D. Ganji, Statistical optimization of microchannel heat sink (MCHS) geometry cooled by different nanofluids using RSM analysis. The European Physical Journal Plus, 130 (2015) 22.