

Process Fault Diagnosis Using Support Vector Machines with a Genetic Algorithm based Parameter Tuning

Mehdi Namdari*, Hooshang Jazayeri-Rad, Seyed-Jalaladdin Hashemi

Department of Instrumentation and Automation, Petroleum University of Technology, Ahwaz, Iran

*Corresponding author: namdari_mehdi@yahoo.com

Received November 25, 2013; Revised December 10, 2013; Accepted December 24, 2013

Abstract Fault diagnosis, centered on pattern recognition techniques employing online measurements of process data, has been studied during the past decades. Amongst those techniques, artificial neural networks classifiers received an enormous attention due to some of their remarkable features. Recently, a new machine learning method based on statistical learning theory known as the Support Vector Machine (SVM) classifier is offered in the pattern recognition field. Support vector machine classifiers were originally used to solve binary classification problems. Subsequently, methods were proposed to apply support vector machine classifier to multiclass problems. Two of these mostly used methods are known as one versus one and one versus all. This paper deals with the application of the above mentioned classifiers for fault diagnosis of a chemical process containing a continuous stirred tank reactor and a heat exchanger. The results show a superior classification performance of the support vector machine versus the selected artificial neural network. In addition, the support vector machine classifier is very sensitive to the proper selection of the training parameters. It is shown that the utilization of genetic algorithm for optimal selection of these parameters is feasible and can help to improve the support vector machine classifier performance.

Keywords: *process fault diagnosis, continuous stirred tank reactor, support vector machine, artificial neural network, SVM Parameter Tuning*

Cite This Article: Mehdi Namdari, Hooshang Jazayeri-Rad, and Seyed-Jalaladdin Hashemi, "Process Fault Diagnosis Using Support Vector Machines with a Genetic Algorithm based Parameter Tuning." *Journal of Automation and Control* 2, no. 1 (2014): 1-7. doi: 10.12691/automation-2-1-1.

1. Introduction

Recently, Process Fault Diagnosis (PFD) systems have been developed to a high degree of quality in modern industry, because the cutting-edge faults diagnosis system has been employed to avoid severe damages. Fault diagnosis objective is to isolate and classify the detected irregularities and also to determine the origin of the observed abnormal status. Fault diagnosis is a crucial step to properly remove the fault. Besides, large volumes of data collected from chemical processes necessitate the development of the data-driven methods to convert them into proper information suitable for many fields such as PFD.

Amongst the numerous approaches that have been employed in PFD, those which utilize the pattern recognition problem are very striking. Because, they do not require any qualitative or quantitative model of the process and solely employ the historical data to design a fault diagnosis model to lay down the current process operating condition. Furthermore, these models permit dealing with the inherent nonlinearities of most chemical processes with no extra cost. In these methods, diverse operating conditions including normal and abnormal ones are considered as patterns. Then, the resulting classifier is used to examine the online measurement data and to convert them to a known class label for

abnormal or normal so that the existing system condition is categorized.

Many methodologies based on the so called pattern recognition techniques have been implemented. Amongst those methods, Artificial Neural Networks (ANN) had an enormous consideration in past years because of some of their remarkable characteristics such as dealing with nonlinearity, noise tolerance and generalization capability [1,2,3,4,5]. ANNs have demonstrated to be good classifiers; however, they require a large number of samples for training, which are not every time essentially obtainable. Also, they infrequently give high generalization error because in the training process they only attempt to enhance categorization performance due to the training data. To resolve this problem, a new machine learning technique relying on statistical learning theory known as the support vector machine is suggested in pattern recognition issues to achieve high generalization capability and to deal with problems with low samples and high input features [6]. It has been shown that the SVM based classifiers are more effective than the formerly reported pattern classifiers and in recent years have been established to be extremely effective in numerous practical applications [7,8,9,10]. However, their applications to process engineering problems are still rare [11,12,13,14]. This paper makes use of SVM for fault diagnosis of a combination of the Continuous a Stirred Tank Reactor (CSTR) and a heat exchanger in a chemical process plant.

SVM was initially developed for binary classification. Several techniques have been proposed for the SVM classifier to be extended to multiclass problems. Amongst them, two methods: One Versus One (OVO) and One Versus All (OVA) are very popular. So in this work a comparison study will be performed amongst three classifiers: the SVM based on OVO (OVO-SVM); the SVM based on OVA (OVA-SVM) and the ANN classifier. A major improvement of the SVM performance can be achieved by the proper selection of the classifier parameters. Initially, a trial and error (T&E) method is used to select the parameters of both of the OVO-SVM and OVA-SVM classifiers. Then, to obtain the optimal parameter values for the SVM classifiers, the Genetic Algorithm (GA), as a well-known optimization solver is used. The result indicated improved classification performance.

2. Methods

2.1. ANN

ANNs were inspired from the study of the human brain which consists of millions of organized neurons. ANNs were developed to imitate the computational structures of these neurons. They contain numerous linked artificial processing neurons known as nodes which are joined together in layers making a network. A node obtains weighted inputs from other nodes or sources, and directs the summation of them to a transfer function known as the activation function. Activation function is a kind of mathematical function. It is typically selected to be a constrained differentiable function such as the sigmoid.

Of all the arrangements of ANNs, the Multi-Layer Perceptron (MLP) neural network is the most prevalent one. As an example, a MLP network with three layers is presented in Figure 1. The circles signify neurons organized in three layers as input, hidden and output. Each hidden neuron is linked to each input and output units. The number of nodes in the input and output layers are determined by the nature of the problem to be unraveled and the number of input and output variables required to define the problem. The number of hidden layers and the nodes within each hidden layer is typically found by the T&E method.

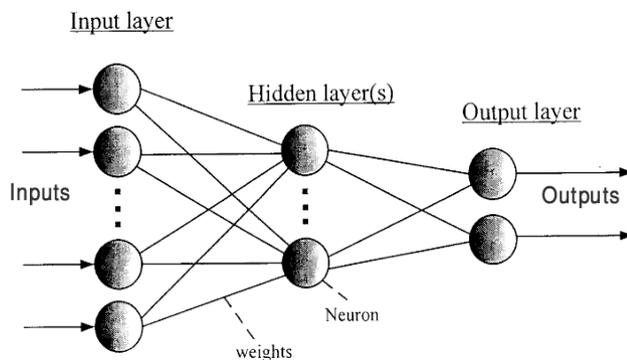


Figure 1. MLP neural network with three layers

Training is used to obtain the values of the signal weights. The network is trained on a collection of data known as the training set, acquired from the process to be modeled. This method is known as the “learning”

procedure. The learning procedure is basically an optimization algorithm. As soon as a network is trained, it can be suitably employed as a model to characterize the system for many different purposes.

2.2. SVM

SVM was initially developed for pattern recognition tasks. Pattern recognition or classification is to label some object into one of the specified categories called classes. Conventional classifiers such as neural networks minimize the training data set error called Empirical Risk Minimization (ERM), while SVM relies on the Structural Risk Minimization (SRM) principle founded on the statistical learning theory, which enhances generalization capabilities.

2.2.1. Two-Class SVM

SVM, in its simple form, is a binary classifier which gives rise to a linear hyper-plane which distinguishes a set of positive examples from a set of negative examples with extreme margin (the margin is outlined by the distance of the hyper-plane to the bordering positive and negative examples as shown in Figure 2). This maximum margin hyper-plane is known as the optimal separating hyper-plane. Because the theories behind the statistical learning endorse that considering maximum margin in training procedure ends up to acquiring better generalization capability [15]. The nearest data points which are employed to define the margin are known as support vectors. The number of support vectors rises with the enormity of the problem.

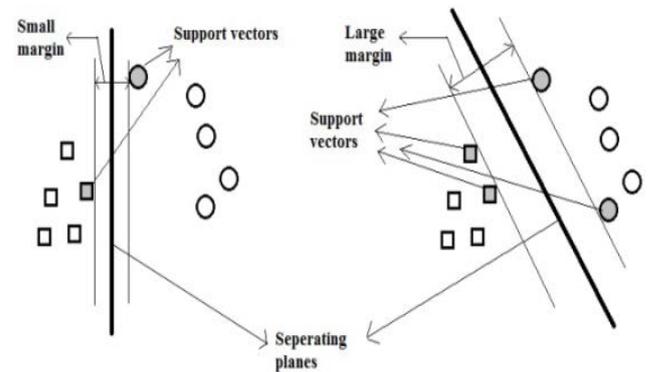


Figure 2. Separating hyper-planes with small and large margins in binary classification

Assume there is a known training sample set $G = \{(x_i, y_i), i=1...M\}$, where M is the number of samples and each sample $x_i \in R^d$ fits into a class by $y_i \in \{+1, -1\}$. In the case of a linearly separable data set, it is possible to separate the given data into the two classes using the hyper-plane:

$$\mathbf{W}^T \cdot \mathbf{x} + b = 0 \quad (1)$$

where \mathbf{W} is an M -dimensional vector and b is a scalar bias term. The vector \mathbf{W} and scalar b are employed to describe the position of the separating hyper-plane. The following decision function, $D(x_i)$, can be employed to categorize input data into either positive class or negative class. So for a known input data x_i :

$$D(x_i) = \mathbf{W}^T \cdot \mathbf{x}_i + b \begin{cases} > 0 \text{ for } y_i = +1 \\ < 0 \text{ for } y_i = -1 \end{cases} \quad (2)$$

It is possible to demonstrate that maximization of the margin can be achieved through minimization of \mathbf{W} . In order to obtain a hyper-plane with larger margin which results in better generalization ability, SVM defines a slack variable ζ_i for each training sample and allows some of them to be misclassified. So the optimal hyper-plane separating the data can be determined as a solution to the following constrained quadratic optimization problem:

$$\text{Minimise } \frac{1}{2} \|\mathbf{W}^2\| + c \sum_{i=1}^M \zeta_i \quad (3)$$

$$\text{Subject to } [\mathbf{W} \cdot \mathbf{x}_i + b] \geq 1 - \zeta_i, i = 1, \dots, M \quad (4)$$

where c is the regularization parameter defined as the trade-off between the margin maximization and classification error minimization. Hence, it can be set to prevent the over-fitting phenomenon.

If the input data are not linearly separable, the original input space is mapped into a high-dimensional space called the feature space. It is then possible to determine a hyper-plane that allows linear separation in the feature space. SVM employs kernels to transform the data into higher dimensions. The most common kernel function is Radial Basis Functions (RBF) which is defined as:

$$K(x, y) = \exp(-\gamma \|x - y\|^2) \quad (5)$$

2.2.2. Multi-Class SVM

The discussion above handles binary classification where the class labels can take only two values: 1 and -1. In practice, however, we find more than two classes for examples in fault diagnosis. Unfortunately there is no unique way to employ SVM for multiclass problems. At this time, there are two general methodologies for multiclass strategies. One is creating and combining several binary classifiers while the other is by directly employing all data in an optimization problem. Two techniques which are used in this paper employ the first approach to tackle the multiclass problem. These are the OVA and OVO methods. The OVA method constructs k binary classifiers where k is the number of classes. Each binary classifier splits the training samples of one class from all other classes. That is why it is known as the one versus all. So in OVA for the k binary SVMs, all training samples of all classes are used. However, the second strategy, OVO, creates $k*(k-1)/2$ binary classifiers where each binary classifier separates the training samples of one class from another and so in training of each binary SVM, training samples of only two classes are employed. Further details of these two strategies and other existing methods for the multiclass SVM can be found in [16].

2.3. GA

The genetic algorithm is a technique for resolving optimization problems that relies on natural selection which is a process motivating biological evolution. The genetic algorithm continually adjusts the population of individual solutions. At each step, the genetic algorithm picks individuals at random from the current population to be parents and uses them to generate the children for the next generation. Over consecutive generations, the population evolves toward an optimal solution. GA can

handle large search spaces effectively, and therefore has less chance to get into local optimal solutions than other algorithms. The genetic algorithm generates three types of children for the next generation:

- Elite children are the individuals in the existing generation with the optimal fitness values. These individuals are routinely carried on to the next generation.
- Crossover children are generated by joining the vectors of a pair of parents.
- Mutation children are generated by producing random variations, or mutations, into a single parent.

Figure 3 illustrates the genetic operations of crossover and mutation. Detailed description of GA can be found in [17].

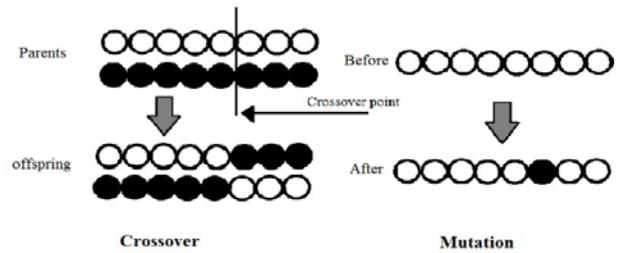


Figure 3. Genetic crossover and mutation operations

3. Case Study

The process used in this paper is shown in Figure 2. This process is made up of a heat exchanger and a CSTR where an irreversible first-order reaction $A \rightarrow B$ occurs. The reaction is catalytic and exothermic. The temperature of the reactor (T_m) is regulated by pumping a part of the reactor outlet stream back to the reactor through the heat exchanger where the temperature (T_R) of the recycle flow (F_R) is cooled by an external flow (F_w). F_R is maintained using a flow controller. The process employs two other feedback controllers (CL and CT) that keep the level of the reactor (L_m) and the temperature of the reactor at the set point temperatures of L_s and T_s , respectively. This process is selected because it reveals the most common features appearing in industrial processes. The fundamental model of the process can be found in [18].

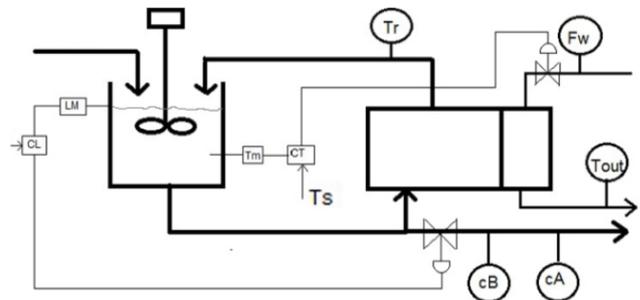


Figure 4. The simulated process

Altogether, five variables (F_w , T_r , T_{out} , c_A , and c_B) are measured from the process (the encircled variables in Figure 4). T_{out} is the output temperature of the coolant fluid; c_A and c_B are the concentrations of the components A and B, respectively. Eight typical fault situations have been chosen for this study. These faults are listed in Table 1. Each fault in the heat exchanger-CSTR system

corresponds to fluctuations in one related faulty variable. For each fault situation, different step changes within the range of 0-10% in the corresponding faulty variable generated different faulty samples. For each faulty case, 100 data samples were generated for the network training and 100 samples were generated for the network testing. Gaussian noise has been added to both of the training and testing data. Also, some outlier samples are added to the generated data.

Table 1. Fault Situations in the Example Process

Faults	Description
Fault 1	Low feed concentration
Fault 2	Low recycle flow
Fault 3	Bias error in the T_2 sensor
Fault 4	Fouled heat exchanger
Fault 5	Deactivated catalyst
Fault 6	Low inlet flow
Fault 7	Low feed temperature
Fault 8	Low cooling water temperature

It is very difficult to comprehend the geometrical properties of different patterns in a 5-dimensional measurement space. Therefore, we need to use a dimensionality reduction technique to visualize the samples in a lower dimensional space. The Principal Component Analysis (PCA) technique is used for this purpose. A detailed description of PCA is beyond the scope of this work. Briefly, PCA reduces the dimension of the data to a number of principal components which explain different proportion of the variance within the original data. Typically, the first three components account for the majority of the variations, and may be used to represent the data in a lower dimension. Figure 5 presents the generated samples for eight fault situations with the first and second principal components. It should be noticed that PCA here is used only for the visualization purpose and geometrical comprehension about different generated faulty samples. The original 5-dimensional patterns will be used to train the classifiers.

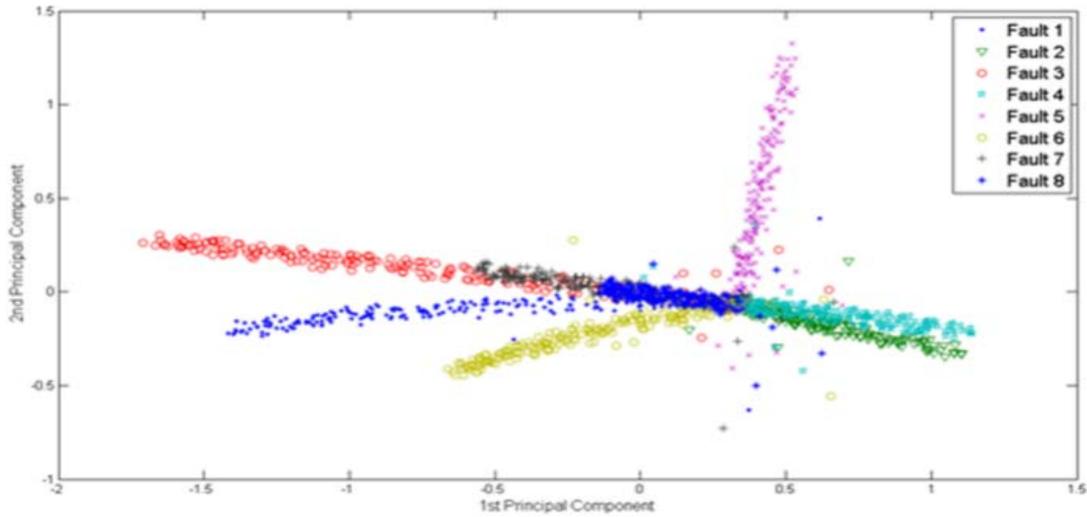


Figure 5. The fault situations of example process presented with the first and second principal components

4. Results and Discussion

4.1. Classification Performance Comparison

For training of both of the ANN and SVM classifiers, training and testing data were normalized within the range [0, 1]. For the ANN a two layers feed-forward MLP network with sigmoid activation function for both hidden and output layers has been chosen. The number of neurons in the output layer is fixed and is equal to the defined number of fault situations which is equal to eight. The number of neurons in the hidden layer should be determined. This highly affects the classification performance of ANNs. So the ANN is trained for different numbers of hidden neurons and then the network with the least average misclassification rate is selected. In the training cycle for each fault situation, its output value is set to 1 and the output values corresponding to other fault situations are set to 0. In testing the network, for an arbitrary input, a particular fault will be detected if its corresponding output value is greater than the values at other outputs.

For training the ANN the Scaled Conjugate Gradient (SCG) algorithm has been used. The MATLAB 7.12

neural network toolbox has been used for this purpose. The network is trained several times with different random initial weight values. The number of training iteration is set to 200. Then, the network with the best classification performance is chosen. The classification performance of the trained ANN is reported in Table 2 and shown in Figure 6.

For training the SVM classifiers first a suitable kernel function should be chosen. Usually the RBF kernel is a reasonable first choice because it only needs two parameters (c , γ) which need to be determined and has fewer numerical difficulties than the polynomial and sigmoid kernels [19]. So, RBF kernel has been selected for this work. The two parameters (c , γ) are determined using the T&E method. For training the SVM classifiers, the Libsvm toolbox developed by Chang and Lin has been used [20]. Libsvm considers a Sequential Minimal Optimization (SMO) type decomposition method for training the SVM classifiers. The classification performances of both of the OVA-SVM and OVO-SVM classifiers are reported in Table 2 and shown in Figure 6.

As seen in Table 2, the magnitudes of average misclassification rates indicate that the classification performances of the OVO-SVM and OVA-SVM classifiers outperform the ANN classifier. Furthermore,

although a fast training technique such as the SCG algorithm has been used to train the ANN classifier, the training rate of each of the OVA-SVM and OVO-SVM classifiers was much faster than the ANN classifier (about three times faster). A major difficulty in using neural network is that it may entrap in local minima during the training cycle. To resolve this problem, the network should be trained with different initial weights. However,

the SVM classifier always converges to a unique solution in the training phase. These major advantages of the SVM classifier over the ANN classifier make it a better tool for pattern recognition and process fault diagnosis tasks. Also, the performance of the OVO-SVM is better than the OVA-SVM with respect to the classification rate. Therefore, the OVO technique is better than the OVA scheme for the multiclass SVM as was reported in [16].

Table 2. Average Misclassification Rates of the Trained Classifiers

Classifier	Parameter	Misclassification Rate (test data) (%)	Misclassification Rate (training data) (%)
ANN	hidden neurons= 6	14	12.625
OVA-SVM	$(c, \gamma) = (2^3, 1)$	12.75	10.875
OVO-SVM	$(c, \gamma) = (2^3, 1)$	11.875	9.875

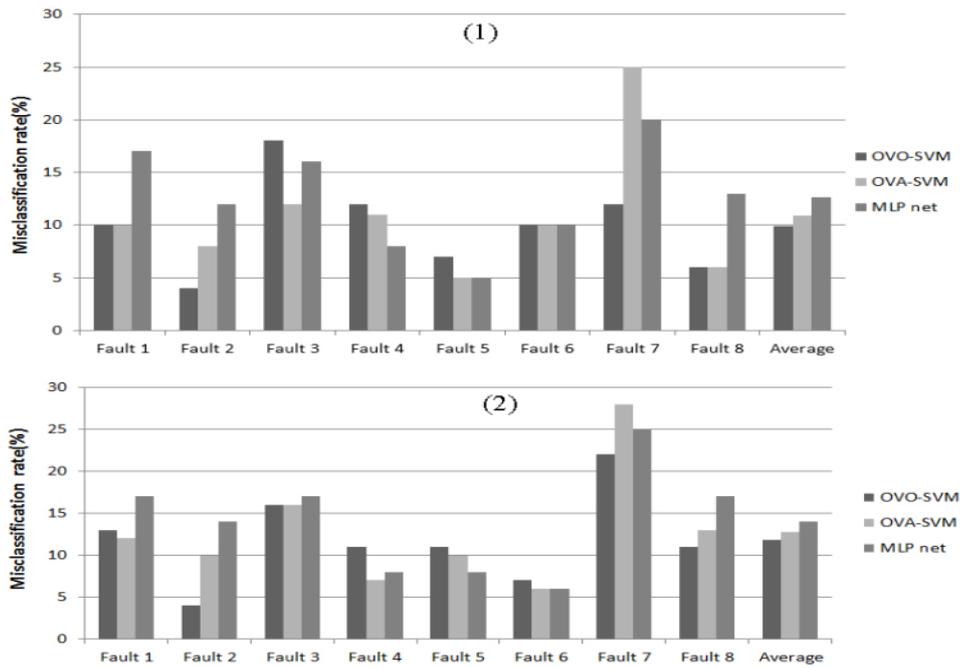


Figure 6. Misclassification rates of different fault situations for the training (1) and test (2) data

4.2. GA-based Parameter Tuning

Classification performance of the SVM classifier to a large extent depends on its regularization parameter c and the kernel function parameters. The common approach to find the optimal values for these hyper-parameters is to use a grid search with cross-validation [19]. In the grid search, various pairs of (c, γ) values are tried and the one with the best cross-validation accuracy is selected. In the ν -fold cross-validation, the training set is divided into ν subsets of equal sizes. Sequentially, one subset is tested using the classifier trained on the remaining $\nu-1$ subsets. Thus, each instance of the whole training set is predicted once and the cross-validation accuracy is the percentage of data which are *incorrectly* classified.

The application of grid search combined with the cross-validation for a wide range of the parameters (c, γ) can be very time consuming and may not result in the optimal hyper-parameters. So in this work, a GA-based parameter tuning method is used and proposed to automatically obtain the optimal hyper-parameters of the SVM classifier. The classifier is trained and validated using the ν -fold cross-validation based on the training parameters proposed by the GA. The classification rate of the training data is the fitness function of the GA. Therefore, the optimum tuning is achieved when a minimum misclassification is

detected by the classifier. This procedure is shown in Figure 7.

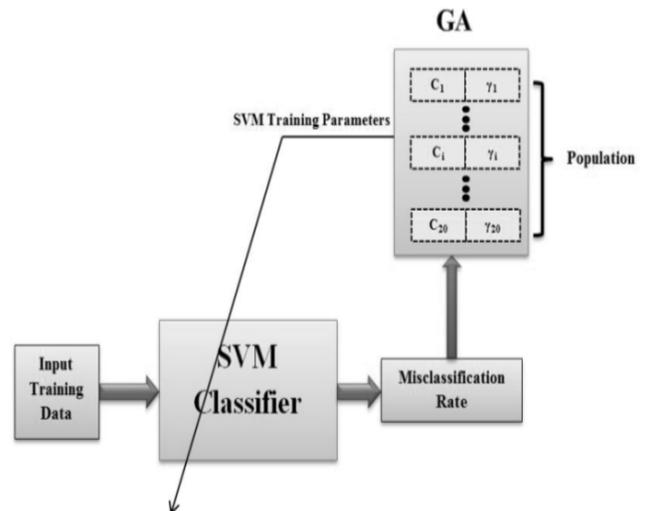


Figure 7. GA-based parameter tuning procedure

For both the OVO and OVA methods, the fitness function is the classification rate of a 10-fold cross-validation procedure and the population size is considered to be equal to 20. The initial range is taken to be within [0,

2] for all individuals. Figure 8 (1) and Figure 8 (2) present the optimization procedures and as seen in each of these figures the fitness function has been minimized under 30 iterations. Table 3 and Figure 9 show the classification performance of both of the OVA and OVO methods using the T&E method of parameter selection and the GA-based

parameter tuning. As seen in Table 3, the classification performances of both of the OVA and OVO methods, when compared against the T&E scheme, are improved using the GA-parameter tuning. These results show the advantage of using the GA for the parameter tuning of the SVM classifier.

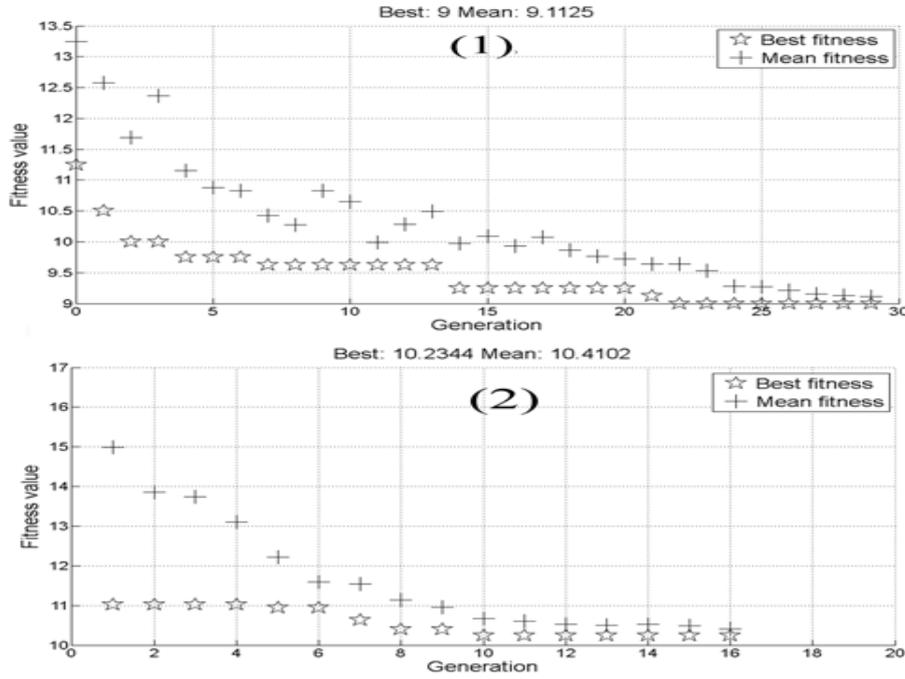


Figure 8. GA-based optimization procedures for the OVO-SVM (1) and OVA-SVM (2)

Table 3. Average Misclassification Rate with the Trial and Error (T&E)- and GA-based Parameter Selections

Classifier	Parameter (c, γ)	Misclassification Rate (testing data) (%)	Misclassification Rate (training data) (%)
T&E-OVA	(2 [^] 3, 1)	12.75	10.875
GA-OVA	(43.9,18.3)	12	3.625
T&E-OVO	(2 [^] 3, 1)	11.875	9.875
GA-OVO	(1.5,103.9)	10.875	4.25

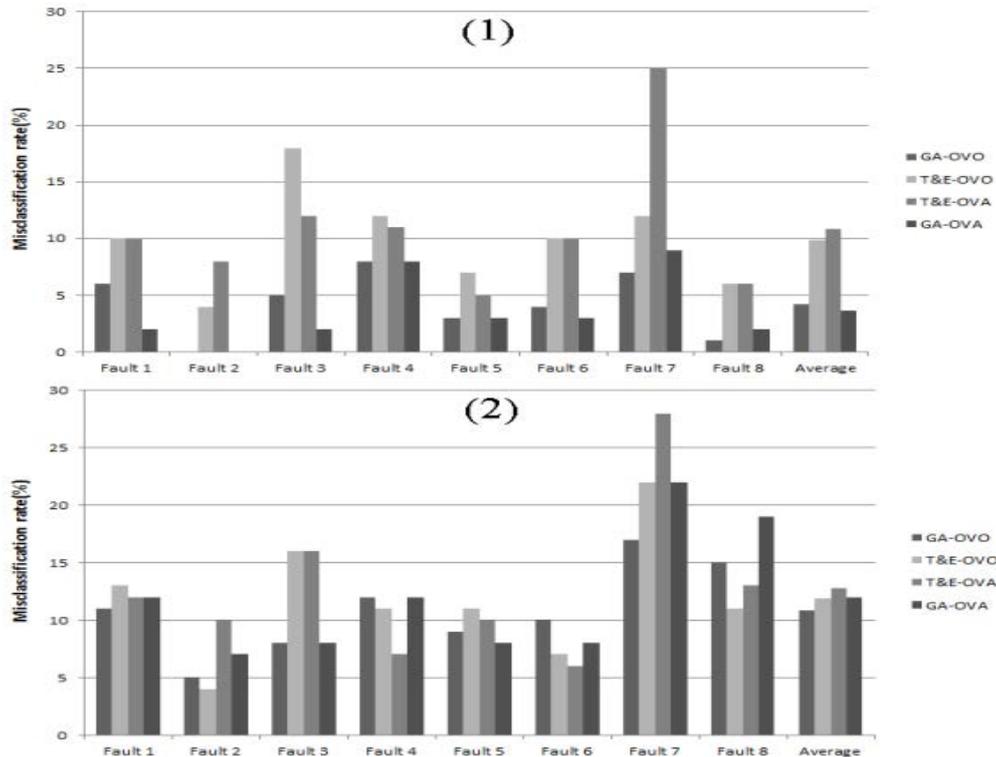


Figure 9. Misclassification rates of different fault situations with trial and error (T&E)- and GA-based parameter selections for the training (1) and test(2) data

5. Conclusions

In this work the SVM and ANN classifiers are applied to the fault diagnosis of a chemical process containing a CSTR and a heat exchanger. As expected, the results show a superior classification performance for the SVM classifier against the ANN classifier. The performance measures were based on the misclassification rates. Also the entrapment in local minima problem for ANNs, sometimes necessitate the training procedure to be repeated several times with different initial weights to achieve acceptable results. In contrast, the objective function formulation in SVM gives a quadratic problem with linear constraint that has a distinctive solution; therefore, training of the SVM classifier always leads to a unique result. These major advantages of the SVM classifier over the ANN classifier make it a superior tool for pattern recognition and process fault diagnosis tasks. In addition, the performances of the OVO-SVM and OVA-SVM approaches in multiclass problems have been compared and the results show that the OVO scheme outperformed the OVA method for the multiclass SVM. Finally, it has been shown that the use of the T&E method for the selection of the training parameters provides an improved classification performance for the SVM classifier against the ANN classifier. In addition, the results show that a GA-based parameter tuning for the SVM classifier gives rise to further enhancement of the classification performance of the SVM classifier.

References

- [1] Watanabe, K., Matsuura, I., Abe, M., Kubota, M. and Himmelblau, D. M., "Incipient fault diagnosis of chemical processes via artificial neural networks," *AIChE Journal*, 35 (11), 1803-1812, 1989.
- [2] Koivo, H. N., "Artificial neural networks in fault diagnosis and control," *Control Engineering Practice*, 2 (1), 89-101, 1994.
- [3] Sharma, R., Singh, K., Singhal, D. and Ghosh, R., "Neural network applications for detecting process faults in packed towers," *Chemical Engineering and Processing: Process Intensification*, 43 (7), 841-847, 2004.
- [4] Behbahani, R. M., Jazayeri-Rad, H. and Hajmirzaee, S., "Fault detection and diagnosis in a sour gas absorption column using neural networks," *Chemical engineering & technology*, 32 (5), 840-845, 2009.
- [5] Eslamloueyan, R., "Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee-Eastman process," *Applied Soft Computing*, 11 (1), 1407-1415, 2011.
- [6] Vapnik, V., *The nature of statistical learning theory*. New York, Springer, 1999.
- [7] Widodo, A. and Yang, B.S. "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, 21 (6), 2560-2574, 2007.
- [8] Tyagi, C. S., "A comparative study of SVM classifiers and artificial neural networks application for rolling element bearing fault diagnosis using wavelet transform preprocessing," *Neuron*, 1, 309-317, 2008.
- [9] Lee, M. C. and To, C., "Comparison of support vector machine and back propagation neural network in evaluating the enterprise financial distress," *International Journal of Artificial Intelligence & Applications*, 1(3), 31-43, 2010.
- [10] Azar, A. and El-Said, S., "Performance analysis of support vector machines classifiers in breast cancer mammography recognition," *Neural Computing and Applications*, 1-15, 2013.
- [11] Chiang, L. H., Kotanchek, M.E. and Kordon, A. K., "Fault diagnosis based on Fisher discriminant analysis and support vector machines," *Computers & chemical engineering*, 28 (8), 1389-1401, 2004.
- [12] Yélamos, I., Graells, M., Puigjaner, L. and Escudero, G., "Simultaneous fault diagnosis in chemical plants using a multilabel approach," *AIChE Journal*, 53 (11), 2871-2884, 2007.
- [13] Mao, Y., Xia, Z., Yin, Z., Sun, Y. and Wan, Z., "Fault diagnosis based on fuzzy support vector machine with parameter tuning and feature selection," *Chinese Journal of Chemical Engineering*, 15(2), 233-239, 2007.
- [14] Yélamos, I., Escudero, G., Graells, M. and Puigjaner, L., "Performance assessment of a novel fault diagnosis system based on support vector machines," *Computers & Chemical Engineering*, 33 (1), 244-255, 2009.
- [15] Abe, S., *Support vector machines for pattern classification*. New York, Springer, 2005.
- [16] Hsu, C. W. and Lin, C. J., "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, 13(2), 415-425, 2002.
- [17] Whitley, D., "A genetic algorithm tutorial," *Statistics and computing*, 4(2), 65-85, 1994.
- [18] Sorsa, T., Koivo, H. N. and Koivisto, H., "Neural networks in process fault diagnosis," *Systems, Man and Cybernetics, IEEE Transactions on*, 21 (4), 815-825, 1991.
- [19] Hsu, C. W., Chang, C.C. and Lin, C. J., "A practical guide to support vector classification," 2010.
- [20] Chang, C.C. and Lin, C. J., "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 2011.