



# **iJRASET**

International Journal For Research in  
Applied Science and Engineering Technology



---

# **INTERNATIONAL JOURNAL FOR RESEARCH**

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume: 5      Issue: X      Month of publication: October 2017**

**DOI: <http://doi.org/10.22214/ijraset.2017.10053>**

**[www.ijraset.com](http://www.ijraset.com)**

**Call:  08813907089**

**E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)**

# Optimum Parameters for Fault Detection in Bioreactor Using Support Vector Machine and Neural Network

Mr. Rahul Shrivastava<sup>1</sup> Dr. Hari Mahalingam<sup>2</sup> Prof. NN Dutta<sup>3</sup>

<sup>1</sup>Department of Chemical Engineering, Jaypee University of Engineering and Technology, AB-road, Raghogarh, Guna (MP), INDIA-473226.

<sup>2</sup>Department of Chemical Engineering National Institute of Technology Karnataka, Surathkal, INDIA-575025.

<sup>3</sup>Department of Chemical Engineering, Jaypee University of Engineering and Technology, AB-road, Raghogarh, Guna (MP), INDIA-473226.

**Abstract:** *Bioprocesses are parameter sensitive, highly non-linear and complex in nature. Apart from close monitoring of the process variables and good control strategy, a robust fault detection and diagnosis (FDD) strategy will help in maintaining the quality of the product by early detection of fault. This paper discusses the application of support vector machines (SVM) and artificial neural networks (ANN) in classifying the normal and faulty conditions in semi-batch bioreactor for penicillin production. The SVM classifier was implemented with three kernel functions viz. linear, polynomial and Gaussian radial basis function (RBF). In case of ANN, a back propagation two layer feed forward neural network was used. The optimum SVM and ANN models were found when the model produced the minimal error with respect to the training, testing and data validation. In case of SVM, it was found that linear kernel is not suitable for bioreactor data, whereas RBF outperformed the polynomial function. Further the performance of SVM was compared with ANN, and ANN outperformed SVM.*

**Keyword:** *fault detection, bioreactor, support vector machine, artificial neural network.*

## I. INTRODUCTION

In order to achieve maximum productivity, to reduce product rejection rate and to satisfy safety and environmental regulations there has been great interest in the development of FDD methods. In past decades, Different methods have been developed to detect and diagnose faults in complex systems. FDD approaches can be roughly divided into two major categories: model based methods and process history based methods. Model based methods make use of the quantitative/qualitative models. Quantitative models are based on first principles of a physical system and qualitative models are based on the available information and knowledge of a physical system [1, 2]. On the other hand, process history based methods require large amount of historical data, which contain the typical trends and previous fault information for effective monitoring methods to be built [3]. Generally, the model-based methods are more accurate, provided that the exact mathematical model is available. However, it is difficult to develop the mathematical model for a process which is complex and highly non-linear in nature. For such processes, process history based methods are found to be more successful than model-based methods. In the last decade the researchers have extensively used process history based methods for studies such as [4-6] for ANN and [7-9] for SVM.

In practice, bioprocesses are parameter sensitive, highly non-linear and complex in nature. In most of the cases, it is difficult or impossible to find rigorous mathematical model of such processes. Therefore, process history based methods can be more useful in case of bioprocesses. In this study, the use of ANN and SVM techniques has been explored in a classical bioreactor problem, namely the production of penicillin. Patil and Kulkarni [10] developed a simple visualization technique to understand the system dynamics in bioreactor for penicillin production. Further, Krishna *et al* [11] studied the parameter interactions for the same process. They selected the bioreactor from the process of penicillin production, because a lot of information is available regarding system dynamics, which can be used to validate the results produced. For simulations, they used PENSIM simulator developed at Chemical Engineering Department of Illinois Institute of Technology, Chicago [12]. This simulator can be used to produce data in both normal and faulty conditions. We have also used this simulator for producing sample data for the purpose of training and testing of SVM and ANN models. The work presented in this paper mainly focuses on the development of optimum SVM and ANN models, which are able to classify normal and faulty conditions in an efficient manner.

The article is organized as follows. The section.2 contains the brief description of SVM and ANN, and proposed method is also representing in this section. Section.3 presents the dataset description. Section.4 is result and discussion section, where the fault classifier were constructed, trained and tested for performance comparison. Finally, conclusion was drawn in section.5.

## II. THEORY & METHODS

### A. Support vector machines (SVM)

SVM were introduced by Bose et al [13]. SVM are based on the Structural Risk Minimization principle from the Statistical Learning Theory [14]. SVM are basically binary classification methods, which separates the negative instances from positive ones by learning a maximum margin linear hyperplane between them.

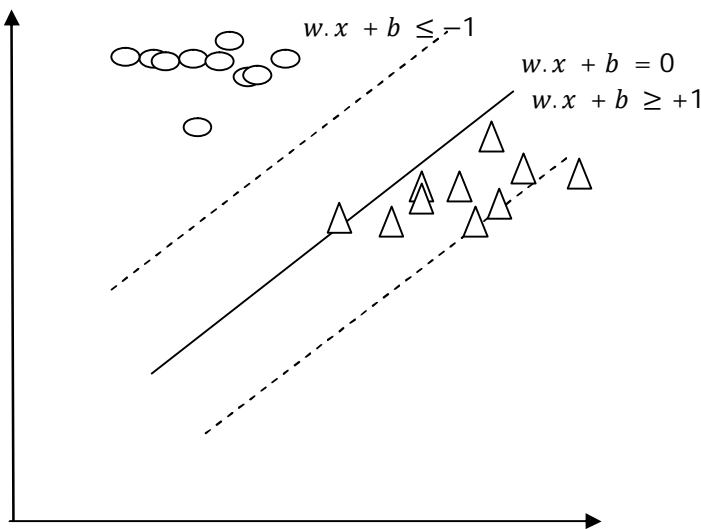


Figure 1. Linearly separable data

See Fig.1, a series of data points for two different classes are shown. Negative and positive classes are represented by ovals and triangles respectively. A linear boundary between the two different classes is placed. Furthermore, SVM orients the boundary in such a way that the distance between the boundary and the nearest data point in each class is maximal. The nearest data points that used to define the margin are called support vectors.

The linear classifier is defined by two elements: a weight vector  $w$  (with one component for each feature), and a bias  $b$ . The classification rule assigns  $+1$ , to a new example  $x$ , when  $f(x) = x \cdot w + b \geq 0$ , and  $-1$  otherwise.

In order to implement support vector machine classifier, margin between the samples has to be maximized, i.e.  $\frac{2}{\|w\|}$ . The formulation of SVM will be as follow

$$\text{Minimize } \frac{1}{2} \|w\|^2 \text{----- (1)}$$

$$\text{Subject to constraint } y_i(w \cdot x_i + b) - 1 \geq 0 \forall i \text{----- (2)}$$

This is primal formulation of linear SVM, which is a convex quadratic optimization problem. Further, this problem can be recast into dual form, by using Lagrange multiplier. One can refer Burges [15] for detailed description of the derivation.

$$\text{Maximize } \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j (x_i \cdot x_j) \text{----- (3)}$$

$$\text{Subject to: } \alpha_i \geq 0 \text{ and } \sum_{i=1}^N y_i \alpha_i = 0 \text{----- (4)}$$

$$\text{Then the vector } w \text{ is defined in terms of } \alpha_i: w = \sum_{i=1}^N \alpha_i y_i x_i \text{----- (5)}$$

Generally, real data sets cannot be separated as cleanly as shown in the Fig.1, instead the data set may contain some points, which will fall on the wrong side of the separating hyperplane or training samples are not linearly separable (Fig.2).

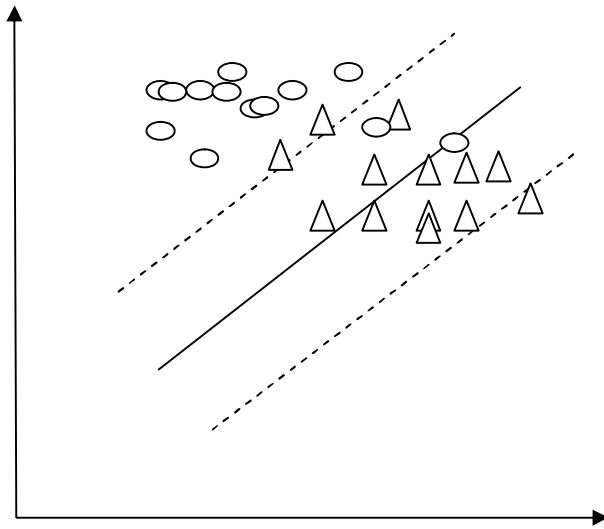


Figure 2. Slightly non-linear data

This problem is resolved by allowing a few anomalous data points to fall on the wrong side of the separating hyperplane. This is done by assigning a positive slack variable to each instance  $\xi_i \geq 0$  in the constraints, which is the distance from the separating hyperplane if an instance is misclassified and 0 otherwise. Further, the problem is converted into the following form

Primal formulation:

$$\text{Minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \text{ ----- (6)}$$

$$\text{Subject to } y_i(w \cdot x_i + b) \geq 1 - \xi_i \text{ for } i = 1 \dots \dots N. \text{ ----- (7)}$$

Dual formulation:

$$\text{Maximize } \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_j \alpha_j (x_i \cdot x_j) \text{ ----- (8)}$$

$$\text{Subject to } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^N \alpha_i y_i = 0 \text{ ----- (9)}$$

The solution is again given by

$$w = \sum_{i=1}^{N_s} \alpha_i y_i x_i \text{ ----- (10)}$$

Where,  $N_s$  is number of support vectors.

Where  $C$  is soft-margin parameter, balancing the classification accuracy and complexity of the decision boundary.

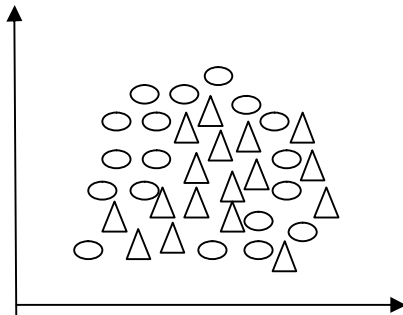


Figure 3. Highly non-linear data

If the training data is highly non-linear and not separable using linear SVM (Fig.3), kernel trick provides a solution to this problem. The kernel functions make SVM work efficiently and implicitly in very high dimensional space. If a good kernel function is smartly selected, the data will become separable in the resulting higher dimensional space. The performance of SVM depends

mainly on the parameters of kernel functions and the soft-margin parameter  $C$ . See Christianini and Shawe-Taylor [16] for details about soft margin and non-linear SVM.

In practice, various kernel functions such as linear, polynomial, radial basis and sigmoid are used.

Linear kernel function:

$$K(x_i, x_j) = x_i \cdot x_j \text{ ----- (11)}$$

Polynomial kernel function:

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^p \text{ ----- (12)}$$

$p$  is order of polynomial. On using high values of  $p$ , there may be a problem of generalization due to over fitting. Value of  $p$  must be selected carefully.

RBF kernel function:

$$K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2) \text{ ----- (13)}$$

Sigmoid kernel function:

$$K(x_i, x_j) = \tanh [a(x_i \cdot x_j) + d] \text{ ----- (14)}$$

In our analysis, we have used linear, polynomial and RBF kernel

**B. Artificial neural network (ANN)**

Several neural network architectures and learning algorithm have been proposed over the years [17]. We are just giving the brief description of feed forward ANN with back propagation algorithm. The ANN can be based on supervised and unsupervised learning. Feed forward network with back propagation belongs to supervised learning category.

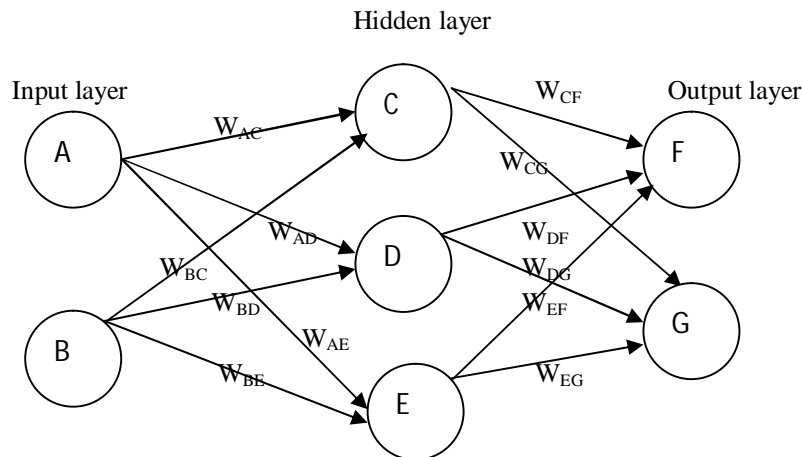


Figure 4. An artificial neural network

Fig.4 is showing the simplest form of feed forward network. In feed forward networks, information flows in one way only i.e. from input to output units, in a feed forward direction. A neural network consists of three layers, i.e. input, hidden and output layer. The process; whereas the output layer consists of output nodes i.e. number of outputs/classes in the process. The hidden layer consists of a certain number of nodes, however, the appropriate numbers of hidden nodes cannot be known in advance. A multilayer network can also be used, which has more than one hidden layer. Fig.5 shows a simple artificial neural network, consisting of a single neuron.

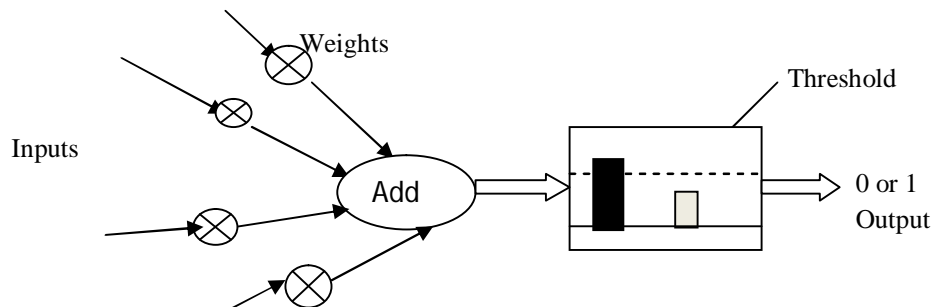


Figure 5. Artificial neural network consist of single neuron



Referring to the Fig.4 and Fig.5, output from the network can be given as follow

$$Y = f\left(\sum W_i X_i - t\right) \text{ --- (15)}$$

Where,  $Y$  is activation result,  $f$  is function of perceptron learning algorithm, whereas  $W$ ,  $X$  and  $t$  are weight vector, input vector and target value respectively.

Between the input, output and hidden layers, there are layers of weights that serve as a connection between them. There are various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to train the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule (supervised learning). In case of supervised learning, network is supplied with given set of input data together with desired set of output data (target values), one for each node, at the output layer. Actual output from the network must be matched with the desired output, if any discrepancy is found, then the weights are updated by prevailing learning rule until actual output from the network does not become equal to the desired output. Another method for training the network is known as unsupervised learning, where output are not available but network learn on its own by discovering and adapting to the structural features in the input patterns.

In case of supervised learning, the difference between actual output and desired output can be minimized by using different methods like least absolute deviation, asymmetric least squares, percentage differences, least fourth powers and sum of the squared errors. Sum of the squared error is most common.

The backpropagation technique is employed as a way of reducing error in the network's classification. The calculated error propagates back through the network for reduction. Inputs are propagated to the first layer of hidden units, whose output is calculated and propagated to the next hidden layer. This process is repeated until the output layer is reached. Each output layer unit calculates the activation, from the sum of weighted inputs from previous layers. The error on the desired output is computed and propagated back to the first hidden layer, where the weight matrix is updated. This process is repeated until the error is minimized as much as possible.

Suppose we are given a training set  $\{(x_1, t_1) \dots (x_r, t_r)\}$ , Consisting of  $r$  ordered pairs of  $n$ - and  $m$ -dimensional vectors, which are called the input and output patterns. Let the primitive functions at each node of the network be continuous and differentiable. The weights are initialized randomly. When the input pattern  $x_i$  from the training set is presented to the network, the network will produce an output  $o_i$ , which is compared with target value  $t_i$ . Our objective is to make  $o_i$  and  $t_i$  identical for  $i = 1 \dots r$ , by using a learning algorithm. More precisely, we want to minimize the error function of the network. The error function is defined as follow

$$E = \frac{1}{2} \sum_{i=1}^r ||o_i - t_i||^2 \text{ --- (16)}$$

Where,  $o$  represent the output from the network and  $t$  represents the actual target for a given input. The gradient of the error function is computed and used to correct the initial weights. Our task is to compute this gradient recursively.  $E$  is actually the summation of square of all the errors, i.e. error for each element of input-output data.

The weights in the network are modified each time to make the error  $E$  as low as possible.  $E$  is a continuous and differentiable function of the weights  $w_1, w_2, \dots, w_l$  in the network. We can thus minimize  $E$  by using an iterative process of gradient descent, for which we need to calculate the gradient

$$\nabla E = \left( \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_l} \right) \text{ --- (17)}$$

Weights are updated by using the following increment

$$\Delta w_i = -\gamma \frac{\partial E}{\partial w_i} \text{ for } i = 1, \dots, l \text{ --- (18)}$$

In our analysis, we have used the most popular form of neural network i.e. two layer feed forward network, which has only one hidden layer.

### III. DATASET DESCRIPTION

There are 18 variables, which are listed and classified as follows (Table 1).

Table 1. Variables involved in the process

1. Aeration rate 2. Agitator power 3. Substrate feed rate 4. Substrate feed temperature 5. Generated heat	Input Variables
6. pH 7. Fermentor temperature	Controlled Variables
8. Acid flow rate 9. Base flow rate 10. Cooling water flow rate 11. Hot water flow rate	Manipulated Variables
12. Substrate concentration 13. Dissolved oxygen concentration 14. Biomass concentration 15. Penicillin concentration 16. Culture volume 17. CO <sub>2</sub> concentration	State Variables
18. Time	Independent variable

Out of the variables listed in Table 1, only 16 were used for analysis. The variables excluded were the time step and hot water flow rate; because time is an independent variable and hot water flow rate has zero value throughout the process.

The PENSIM simulator provides for introduction of two fault types like step change and ramp change, which can be given to three variables namely aeration rate, agitator power and substrate feed rate. The first simulation was done for normal operation with default settings and then for four different faulty conditions by providing a step change of magnitude  $\pm 10\%$  to aeration rate and agitator power. If anyone of these two variables goes beyond  $\pm 10\%$  of their normal value, it indicates a fault. See Table 2 for the details of faults employed in this study.

Table 2. Summary of five simulations

S.No.	Faulty variable	Type of fault	Magnitude
1.	Normal operation	No	No
2.	Aeration rate	Step change	+10%
3.	Aeration rate	Step change	-10%
4.	Agitator power	Step change	+10%
5.	Agitator power	Step change	-10%

All the simulations were done for 150 h with a time interval of 0.25 h, thus generating 600 samples for normal and each faulty condition. A total of 3000 samples were generated and used for analysis, out of which 600 belongs to the class of normal operation and rest belong to the class of faulty condition. In addition to these 3000 samples, 1123 more samples were generated randomly for the normal and faulty conditions separately for testing purposes.

#### IV. RESULTS AND DISCUSSION

We have implemented SVM and ANN on the bioreactor data to develop an efficient FDD system, which is able to detect that, whether the system is operating under normal or faulty condition. In case of SVM, bioreactor data was implemented with linear, polynomial and RBF kernel functions. After that feed forward ANN with backpropagation was used. The optimum SVM and ANN models were found when the model produced the minimal error with respect to the training, testing and data validation. We have also observed the training time in each case. For generating an optimum model, it is required to tune the parameters in each case. The parameter  $C$  is common in each case of SVM. In case of polynomial and RBF kernel functions, there is an additional parameter i.e., order of polynomial and  $\sigma$  respectively. In case of polynomial kernel, only third order polynomial was used. Sequential

minimal optimization technique was used to find the optimal separating hyperplane. All the SVM models were trained using 5-fold cross validation. Search for the best parameters values was done for the following sets of values:  $C \in \{1E-05, \dots, 1E+05\}$  in each case and  $\sigma \in \{1E-04, \dots, 1E+04\}$  for RBF kernel. The same dataset was studied with a feed forward ANN with back propagation. All the results were produced using MATLAB.

Table 3. Linear kernel

C	Misclassification error with training samples (%)	Time (s)	Misclassification error with testing samples (%)
1.00E-05	36.7	5.1094	33.03
1.00E-04	33.73	5.4531	34.82
1.00E-03	35.33	5.1875	37.67
1.00E-02	45.7	5.8125	55.03
1.00E-01	46.87	9.1719	42.74
1	46.47	25.1719	56.09
1.00E+01	47.47	402.2188	55.65
1.00E+02		not converging	
1.00E+03		not converging	
1.00E+04		not converging	
1.00E+05		not converging	

From Table 3, it can be observed that the bioreactor data is not separable by using linear SVM. For any value of  $C$ , either the algorithm is not converging or if it converges, the misclassification error with training and testing samples is too high.

Table 4. Polynomial kernel

C	Misclassification error with training samples (%)	Time (s)	Misclassification error with testing samples (%)
1.00E-05	30.6	7.1875	37.49
1.00E-04	15.2	12.2344	40.87
1.00E-03	0	223.6719	6.77
1.00E-02	0	409.2656	5.70
1.00E-01	0	452.3125	5.70
1	0	504.3438	5.61
1.00E+01	0	514.7031	5.43
1.00E+02	0	527.7031	5.43
1.00E+03	0	500.75	5.43
1.00E+04	0	463.8281	5.52
1.00E+05	0	476.5625	5.25
1.00E+100	0	491.2031	5.06

Table 4 presents the results for the case of polynomial kernel function. This indicates that at  $C = 1.00E-05$  &  $1.00E-04$ , error with testing samples is 37.49% & 40.87% respectively. However from  $C = 1.00E-03$  onwards, the error with training samples is nil and



misclassification error with testing samples is also very low, in the range of 5.06% to 6.77%. The training time is in the range from 7 to 527 seconds.

Table 5. RBF kernel

C	$\sigma$	Misclassification error with training samples (%)	Time (s)	Misclassification error with testing samples (%)
1.00E-01	1	0.77	10.9375	6.14
1	1	0.17	13.3281	5.7
1.00E+01	1	0.1	13.5	5.61
1.00E+02	1	0.23	13.6094	5.88
1.00E+03	1	0.1	13.7656	5.61
1.00E+04	1	0.13	12.9844	5.7
1.00E+05	1	0.27	13.7969	5.61
1.00E+01	1.00E+01	0.13	16.25	3.65
1.00E+02	1.00E+01	0	80.5625	1.17
1.00E+03	1.00E+01	0	118.7344	0
1.00E+04	1.00E+01	0	90.9219	0
1.00E+05	1.00E+01	0	76.5469	0.36
1.00E+05	1.00E+02	0.27	905.3906	2.76

Table 5 presents the results for RBF kernel. The best combinations of  $\sigma$  and C are mentioned, rests are provided in the appendix. At some of the combinations of  $\sigma$  and C values, the SVM models were producing nil misclassification error with training and testing samples. Training time is also low as compared to polynomial function.

Table 6. Feed forward ANN with back propagation

No. of neurons	Time (s)	Misclassification error with training samples (%)	Misclassification error with testing samples (%)
10	4	0	0
15	5	0	0
20	8	0	0
25	6	0	0
50	10	0	0
100	24	0	0

See Table 6 for the performance of feed forward ANN with backpropagation. A 2-layer feed forward neural network with sigmoid hidden and output nodes was used. Mean squared error was used as a performance function. Input layer consists of 16 nodes, because there are 16 variables in bioreactor. Second layer is actually the hidden layer, where the number of nodes can be varied by the user. The output from the network will be either 0 (no-fault) or 1 (fault). The network performance was observed by changing the number of neurons in the hidden layer. The network was performing well with any value of number of neurons with very little time required for training and nil misclassification error with training and testing samples.

### V. CONCLUSION

In case of SVM, it is evident that linear kernel is not appropriate for bioreactor data. However, polynomial and RBF kernel functions are able to classify normal and faulty conditions in bioreactor, but RBF outperformed polynomial kernel function. So, the parameter tuning and Choice of kernel function is crucial for efficient classification.

On comparing the results of SVM with feed forward ANN with back propagation, it is clear that ANN has outperformed in terms of training time and misclassification error with training and testing samples. The reason for robust performance of ANN over SVM is

a result of large amount of samples was used for training. ANN typically performs well, if number of classes to be separated is small and large amount of samples are available for training.

**REFERENCES**

- [1] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S. N. Kavuri, "A review of process fault detection and diagnosis Part I: Quantitative model-based methods", Computers and Chemical Engineering, vol. 27, pp. 293-311, 2003a.
- [2] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, "A review of process fault detection and diagnosis Part-II: Qualitative models and search strategies", Computers & chemical engineering, vol. 27, pp. 313-326, 2003b.
- [3] V. Venkatasubramanian, R. Rengaswamy, K. Yin, S. N. Kavuri, "A review of process fault detection and diagnosis Part III: Process history based methods", Computers & chemical engineering, vol. 27, pp. 327-346, 2003c.
- [4] D. Ruiz, J. Nougues-Maria, Z. Calderon, A. Espuna, L. Puigjaner, "Neural network based framework for fault diagnosis in batch chemical plants", Computers and Chemical Engineering, vol. 24, pp. 777-784, 2000.
- [5] Y. Zhou, J. Hahn, M. S. Mannan, "Fault detection and classification in chemical processes based on neural networks with feature extraction", ISA Transactions, vol. 42, pp. 651-664, 2003.
- [6] S. Rajakarunakaran, P. Venkumar, D. Devaraj, K. S. P. Rao, "Artificial neural network approach for fault detection in rotary system", Applied Soft Computing, vol. 8, pp. 740-748, 2008.
- [7] A. Kulkarni, V. K. Jayaraman, B. D. Kulkarni, "Knowledge incorporated support vector machines to detect faults in Tennessee Eastman Process", Computers and Chemical Engineering, vol. 29, pp. 2128-2133, 2005.
- [8] I. Yélamos, G. Escudero, M. Graells, L. Puigjaner, "Fault diagnosis based on support vector machines and systematic comparison to existing approaches", Computers & Chemical Engineering, vol. 21, pp. 1209-1214, 2006.
- [9] I. Yélamos, G. Escudero, M. Graells, L. Puigjaner, "Simultaneous fault diagnosis in chemical plants using support Vector Machines", Computers & Chemical Engineering, vol. 24, pp. 1253-1258, 2007.
- [10] K. R. Patil, A. J. Kulkarni, "A Simple Visualization Techniqueto Understand the System Dynamics in Bioreactors", Biotechnology Progress, vol. 23, pp. 1101-1105, 2007.
- [11] R. Krishna, S. Nanda, A. Kulkarni, S. Patil, "A partial Granger causality based method for analysis of parameter interactions in bioreactors" Computers and Chemical Engineering, vol. 35, pp. 121-126, 2011.
- [12] G. Birol, C. Undey, A. Cinar, "A modular simulation package for fed batch fermentation: Penicillin production. Computers & Chemical Engineering", vol. 26(11), pp. 1553-1565, 2002.
- [13] B. Boser, I. Guyon, V. Vapnik, "A training algorithm for optimal margin classifiers", In Proceedings of fifth annual workshop on computational learning theory, COLT, New York, USA, pp. 144-152, 1992..
- [14] V. N. Vapnik, "Statistical learning theory", John Wiley, 1998.
- [15] C. J. C. Burges, "A tutorial on support vector machines for pattern identification" Daa Mining and Knowledge Discovery, vol. 2, pp. 121-167, 1998.
- [16] N. Christianini, J. Shawe-Taylor, "An introduction to support vector machines", Cambridge University Press, 2000.
- [17] J. A. Anderson, "An introduction to neural networks", MIT Press, 1995.

**APPENDEX**

Remaining results for RBF kernel from Table 5.

C	$\sigma$	Percent error with training samples	Time (s)	Percent error with testing samples
1.00E-05	1	4.7	10.0156	11.75
1.00E-04	1	4.87	10.4063	16.21
1.00E-03	1	4.37	10.5156	12.73
1.00E-02	1	4.67	10.3125	14.16
1.00E-05	1.00E+01	34.73	8.9844	35.17
1.00E-04	1.00E+01	33.7	8.625	31.79
1.00E-03	1.00E+01	33.47	9.0781	31.88
1.00E-02	1.00E+01	34.33	9.0625	39.63
1.00E-01	1.00E+01	40.4	8.7344	33.40
1	1.00E+01	39.7	8.8906	24.13
1.00E-05	1.00E+02	34.53	8.6094	33.21
1.00E-04	1.00E+02	37.23	8.5938	32.77
1.00E-03	1.00E+02	35.93	8.5469	33.93

1.00E-02	1.00E+02	34.47	9.125	35.35
1.00E-01	1.00E+02	39.5	8.8281	33.21
1	1.00E+02	35.27	9.0625	32.24
1.00E+01	1.00E+02	34.83	9.2344	32.32
1.00E+02	1.00E+02	46.73	9.7031	37.49
1.00E+03	1.00E+02	46.7	12.2188	49.42
1.00E+04	1.00E+02	41.7	38.5	39.54
1.00E-05	1.00E+03	33.47	9.0781	38.47
1.00E-04	1.00E+03	35.63	8.8281	37.93
1.00E-03	1.00E+03	34.97	9.2813	40.34
1.00E-02	1.00E+03	34.47	8.75	32.95
1.00E-01	1.00E+03	34.53	8.5938	34.64
1	1.00E+03	34.53	8.6719	36.76
1.00E+01	1.00E+03	34.83	8.8125	33.04
1.00E+02	1.00E+03	34.23	8.9063	35.62
1.00E+03	1.00E+03	34.9	8.8281	36.24
1.00E+04	1.00E+03	45.4	9.2188	41.41
1.00E+05	1.00E+03	47.27	12.5625	55.83
1.00E-05	1.00E+04	34.33	8.8438	35.08
1.00E-04	1.00E+04	34.37	8.7188	33.75
1.00E-03	1.00E+04	33.97	8.6719	34.46
1.00E-02	1.00E+04	35.77	9.0469	35.62
1.00E-01	1.00E+04	35.07	8.8438	34.73
1	1.00E+04	34.13	8.7344	49.69
1.00E+01	1.00E+04	33.7	9.0469	47.55
1.00E+02	1.00E+04	34.43	8.6094	32.24
1.00E+03	1.00E+04	34.73	9.0938	33.93
1.00E+04	1.00E+04	34	8.9375	34.55
1.00E+05	1.00E+04	38.17	9.0625	43.37
1.00E-05	1.00E-01	17.9	10.7031	42.03
1.00E-04	1.00E-01	17.57	10.8594	42.03
1.00E-03	1.00E-01	17.57	11.1875	42.03
1.00E-02	1.00E-01	17.47	10.9531	42.30
1.00E-01	1.00E-01	17.6	11	42.30
1	1.00E-01	10.3	30.6719	39.27
1.00E+01	1.00E-01	10.1	30.9688	39.27
1.00E+02	1.00E-01	10.5	28.7344	40.07
1.00E+03	1.00E-01	10.53	28.3906	38.74
1.00E+04	1.00E-01	10	27.2969	39
1.00E+05	1.00E-01	9.97	29.2969	39.27
1.00E-05	1.00E-02	19.97	9.2031	42.39
1.00E-04	1.00E-02	20.17	8.8438	42.48



1.00E-03	1.00E-02	20.2	8.8281	42.48
1.00E-02	1.00E-02	20.07	9.0469	42.48
1.00E-01	1.00E-02	19.97	9.1563	42.39
1	1.00E-02	19.97	28.4219	42.12
1.00E+01	1.00E-02	20.13	31.25	42.21
1.00E+02	1.00E-02	19.97	29.3281	42.12
1.00E+03	1.00E-02	19.97	30.0156	42.12
1.00E+04	1.00E-02	19.97	30.4531	42.12
1.00E+05	1.00E-02	19.97	31.2813	42.12



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)