IJCTA, 8(3), 2015, pp. 1121-1128
© International Science Press

Data Driven Approach to Monitoring and Fault Detection in Process Control Plants

[1]Linda Varghese, [2]Dr. V.I.George, [3]Dr. Krishnamoorthi Makkithaya, and [4]Abhishek Kumar

Abstract: Detecting faults and monitoring is a very important activity in process control plants for increasing the efficiency of the plant. Data driven approach is particularly helpful for fault detection, if the underlying mathematical model of the process is very complex. It can be used to create automatic systems which accurately predict whether the operating condition of the plant is normal or faulty. This paper compares different supervised learning algorithms, in order to detect fault in process control plant. The algorithms are tested in Matlab environment. Finally, all the models give satisfactory accuracy while detecting two different types of faults as well as normal operating condition.

Keywords: Fault detection, learning from examples, data mining, prediction, Matlab, supervised learning

1. INTRODUCTION

Monitoring is a non-stop activity undertaken in real-time to determine the conditions of a physical system, by recording information, by trying to recognize and gauge behavior anomalies. Data driven approach towards monitoring and fault detection has seen many advances in recent past. We use various data mining techniques on the process control data to train the fault detection system.

The area of data mining is vast and quite popular across several disciplines, such as bio-informatics, data mining, machine learning, applied statistics, image processing, economics etc. A great deal of literature in the form of books and journal articles can be found regarding this field. The most important work done in the field of pattern recognition till date is explained in the papers [1] - [4]. Richard O. Duda et. al [3] discusses in-depth mathematical discussions on regression techniques, decision tree algorithms and neural networks.

Several data mining techniques can be used for fault detection in different industrial domains. Xindong Wu et.al [5] have published a survey paper describing the many important algorithms that are mostly used in the areas of data mining. Many important algorithms such as k-means, support vector machines, CART, Naïve Bayes etc. are discussed. Wei-Yin Loh [6] describes the performance of different decision tree algorithms for classification and regression. The node impurity index and the node split methods of Classification and Regression Tree (CART) algorithm is discussed. Ryan Rifkin et.al [7] have studied the area of multiclass classification and have shown that one-vs.-all scheme used in multiclass classification is as accurate as any other approach used in classification. Decision trees like CHAID,C&R,QUEST,C4.5 etc and prediction algorithms like Neural Network[27], Bayesian, Logistic Regression, SVM etc are compared and discussed in [8][9][10][11][15][21] for their accuracy in identifying faults pertaining to different areas. In order to obtain enhanced classification performance to identify faults, Genetic algorithm[20], neuro fuzzy approach[13] and K-NN[23] could also be used effectively. Kalman filtering and hybrid neuro-

1. Department of Computer Applications, MIT, Manipal. linda.varghese@manipal.edu
2. Department of Instrumentation and Control Engineering, MIT, Manipal. vi.george@manipal.edu
3. Department of Computer Science Engineering, MIT, Manipal. k.moorthi@manipal.edu

fuzzy techniques is proposed by A. Khoukhi et.al [12] for fault detection and classification. Efficient fault detection also finds its application in various semiconductor industries to aid in improved accuracies[14] [16]. The technique called SVM is widely used in chemical industries[17] and thermal power plants[25] in order to predict faults in turbines. Binoy B. Nair Et.al[18] have applied different feature selection techniques to Naïve Bayes classifier to help in prediction of faults in control valves. Yogendra Kumar Jain et. al [24][26] have proposed machine learning application in intrusion detection. Self Organizing Map [29]is used in applications with highly overlapped classes for identification and detection of faults. Elyas Rakshani et. al [19] used artificial neural network for identify and predicting faults in boiler's burner system of power plant. A.A. Shahrjooi Haghighi et. al[15] studied different classification techniques for identifying faults in software. Libo Bie et. al[22] proposed the multiblock principal component analysis (MBPCA) for identifying faults and diagnosing in a continuous process.

2. BACKGROUND THEORY

Data Mining is normally divided into steps of data pre-processing and supervised learning. The process available in data mining used to convert raw data into a meaningful format is known as data pre-processing. As data in real-world would often be not complete, consistent and also be lacking in certain behavior or trend, therefore the chances to have lot of errors can't be ruled out. The final training set is the result of data pre-processing. Min-max normalization is used to for pre-processing of data.

Supervised learning is a method to model the input-output relationship of a given training set. Formally, given a training set, a supervised leaning method learns a function $f: X \to Y$ so that $f(x)$gives a good approximation for Y. Here X and Y denotes the input and output (target) values respectively. The combination of both the input and the output values gives us the training data set. Once the representative function is found, it can be used to find the target values of novel samples.

When the target variable (Y) is continuous, the learning problem is called a classification problem. In all the methods of classification described here, the total set of data has to be broken into training set and test set in the ratio 3:1. This is done so that we can train our model the training set, and then the model assessment is done using the test set.

2.1 Data Pre-Processing: Min-Max Normalization

Data Mining cannot be performed on raw data. It is because the attributes contained in the datasets are on completely different scales, since greater importance can be given to attributes with large ranges than those with small ranges. So, we need to normalize the data on the basis of equal scales for all the attributes. Min-max normalization is used for this purpose. It is given by

$$v' = \frac{v(i,j) - \min(i)}{\max(i) - \min(i)} (new\_max_b - new\_min_b) + new\_min_b \tag{1}$$

Here, the input matrix is scaled between new_min_b and new_max_b. min $_b$ and max_b are the minimum and maximum values of a specific attribute respectively. The scale of $(0,1)$ is used for min-max normalization.

2.2 Linear Regression

Linear regression is a statistical method in which the value of a variable is predicted by modeling its relationship (linearly) with another variable (or a group of variables). The variable to be predicted is called the predicted or output variable, while the variable(s) which we are basing our prediction on is called the predictor. When more than one predictor variables are present, the method is called multivariate linear regression.

Linear regression can be used for classification by providing discrete values to the target variable(Y). Least squares approach is a very popular method for fitting a regression line. In this method, sum of the squares of the vertical deviations from each data points to the line is minimized. In simple words, the aim is to reduce the error of squared function to the negligible limits. Summation of the squares of the error ensures that negative and positive error values do not cancel each other. Given a dataset $X = \{x_{i1}, x_{i2}, \dots, x_{im}\}_{i=1}^{i=n}$ representing the input values and $Y = \{y_i\}_{i=1}^{i=n}$ representing the respective target values, we need to find a mapping from X to Y. In linear regression, it is assumed that this mapping is linear in nature. So, we need to find a set of weights $\theta = \{\theta_i\}_{i=1}^{i=m}$ which gives the nearest approximation of the mapping. Therefore the linear function can be represented in the matrix form

$$Y = X\theta \tag{2}$$

In the above equation, X is $n \times m$ matrix; Y is $n \times 1$ matrix; and θ is an $m \times 1$ matrix. The error function e is defined as

$$e = X\theta - Y \tag{3}$$

Now, we need to minimize the error squared function. Therefore, we define the cost function

$$C = e^2 = (X\theta - Y)^T(X\theta - Y) \tag{4}$$

To minimize the cost function, gradient descent algorithm is used. Gradient descent is based on the observation that if a multivariable function, then f(x) is defined and differentiable in the neighborhood of a point a, then f(x) decreases fastest if one goes from a in the direction of the negative gradient of f at a, $-\nabla f(a)$. It follows that, if $b = a - \alpha\nabla f(a)$, for α small enough, then $f(a) \geq f(b)$. With this observation in mind, one starts with a guess $x_0, x_1, x_2, \dots$ such that

$$x_{n+1} = x_n - \alpha_n \nabla f(x_n), n \geq 0 \tag{5}$$

We have

$$f(x_0) \geq f(x_1) \geq f(x_2) \geq \cdots$$

so hopefully the sequence (x_n) converges to the desired local minimum. We can permit change of the value of the step size (or learning rate) α at every iteration. By allowing some hypothesis on the function f, we can guarantee convergence to a local minimum.

The gradient descent algorithm uses the gradient ∇ of the cost function, i.e. the derivative of the cost function w.r.t. θ. In multivariate linear regression, θ is a vector, containing partial derivative of the cost function C computed w.r.t. each value of θ. Therefore

$$\nabla f(x) = \{\frac{df(x)}{d\theta_1}, \frac{df(x)}{d\theta_2}, \dots, \frac{df(x)}{d\theta_m}\} \tag{6}$$

Thus the gradient of the cost function is found

$$\nabla C = \frac{dC}{d\theta} = X^T(X\theta - Y) \tag{7}$$

The weight matrix θ is then calculated iteratively

$$\theta_{k+1} = \theta_k - \alpha\nabla C \tag{8}$$

where α is the learning rate,

∇C is the gradient of the cost function

the initial value of θ is taken as $\theta_0 = 0$

As a result of gradient descent, the weight matrix θ is obtained such that the error squared function is minimized. Now, θ is used to obtain the output values of the test dataset. Let the test dataset be a $p \times q$ matrix

$$\text{test} = \left\{\text{test}_{i0}, \text{test}_{i1}, \dots, \text{test}_{iq}\right\}_{i=1}^{i=p}$$

The corresponding target values for test dataset is a $p \times 1$ matrix

$$Y_t = \left\{Y_{ti1}, Y_{ti2}, \dots, Y_{tip}\right\}_{i=1}^{i=p}$$

Once the weight matrix θ is obtained, output can be obtained by multiplying θ with the input X

$$Y = X\theta$$

This value of Y is then compared to the target values provided already as part of training dataset Y_t to get the accuracy of the model.

2.3 Logistic Regression

Instead of assuming a linear relationship between input and output values, logistic regression uses the sigmoid (or logistic) function to map input and output vectors. The sigmoid function is defined as

$$\mathrm{sigmoid}(z) = \frac{1}{1+e^{-z}} \tag{9}$$

As in linear regression, we based our hypotheses on the linear relationship of input and output; in logistic regression, the hypothesis is based on the sigmoidal input-output relationship. In the beginning, we define the logistic regression for classification of two classes. Then, multiple classes are taken into account using one-vs.-all classification method.

The notation used is similar to that used in linear regression, with slight changes. Given a dataset $X = \{x_{i1}, x_{i2}, \dots, x_{im}\}_{i=1}^{i=n}$ representing the input values and $Y = \{y_i\}_{i=1}^{i=m}$ representing the respective target values, we need to find a mapping from $X \rightarrow Y$ such that it is sigmoidal in nature. For this, we need to find a vector of weights$\theta = \{\theta_i\}_{i=1}^{i=m}$. The relationship can be written in matrix form

$$h_\theta(X) = \frac{1}{1+e^{-X\theta}} \tag{10}$$

In the above equation, X is $n \times m$ matrix; θ is $m \times 1$ matrix; and $h_\theta(X)$ is $n \times 1$ matrix. Now, we need to define a cost function which penalizes the model for every misclassification made. The cost function is assumed to be

$$C(h_\theta(x)|y = \begin{cases} -\log\big(h_\theta(x)\big) & \text{for } y = 1 \\ -\log\big(1 - h_\theta(x)\big) & \text{for } y = 0 \end{cases} \tag{11}$$

In the above equation, it can be observed that, for $y = 1$, if $h_\theta(x) = 1$, then cost $= 0$; but for $y = 1$, as $h_\theta(x) \rightarrow 0$, cost $\rightarrow \infty$. Similarly, for $y = 0$, if $h_\theta(x) = 0$, then cost $= 0$; but for $y = 0$, as $h_\theta(x) \rightarrow 1$, cost $\rightarrow \infty$. Therefore, the cost function chosen puts a very large penalty for every misclassified instance; and a zero penalty for correctly classified instance. Since the values of $h_\theta(x)$ and y are already known, only the weight vector θ needs to be found. Thus the cost function can be assumed to be a function of θ. Minimizing the value of the above cost function will give us the logistic model in the form of weight vector θ.

Eq. 10 can be combined to give a single cost function

$$C(h_\theta|y) = -y\log\big(h_\theta(x)\big) - (1-y)\log\big(1 - h_\theta(x)\big) \tag{12}$$

For the whole dataset, the cost function is

$$C(\theta) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log\big(h_\theta(x_i)\big) + (1-y_i)\log\big(1 - h_\theta(x_i)\big) \tag{13}$$

Eq. 13 can be written in matrix form

$$C(\theta) = -\frac{1}{n}\big(Y^T \log\big(h_\theta(X)\big) + (I-Y)^T \log\big(I - h_\theta(X)\big) \tag{14}$$

In the above equation, Y^T represents the transpose of matrix Y and I denotes the identity matrix.

We need to minimize eq. 14 to find the weights to represent the model. Minimization of cost function is done using the gradient descent approach as described in the previous section.

The gradient of the cost function is

$$\nabla C = x\big(h_\theta(x) - y\big) \tag{15}$$

In matrix form, the above equation can be written as

$$\nabla C = \frac{1}{n}X^T\big(h_\theta(X) - Y\big) \tag{16}$$

∇C is $m \times 1$ matrix containing the partial derivatives of the cost function w.r.t. the weight θ.

Using gradient descent, the weight matrix θ is calculated iteratively. As a result of this approach, a vector of weights θ is obtained such that it minimizes the misclassification cost C.

Note that the classifier obtained from the above approach can be trained for only two classes. But it can be extended for more than two classes. Suppose, we have three classes, represented by $\{0,1,2\}$. We could use two set of classification. In the first case, class 0 is assumed to be one class and classes $\{1,2\}$ is assumed to be of another class. In the second case, classes $\{0,1\}$ is assumed to be of one class and class 2 is assumed to be of another class. After combining the above two classifiers, a single classifier can be constructed such that it classifies examples from three classes. For this, two sets of weight vector θ are needed. Similarly for a c class problem, $(c-1)$ weight vectors are needed. After the model is trained, it is needed to be tested with test set as described in sec. 3.1.

2.4 Classification and Regression Tree (CART)

Data responses are predicted using classification and regression trees. In order to correctly calculate and assume a response, we need to track the decisions in a tree beginning from the root node making the way down to a leaf node. It is worth noting here that the response is always contained within the leaf node. The responses obtained from classification trees are always nominal, while those from regression trees are always numeric.

Algorithm

1. Initialize with all categories of input data, and check all combinations of binary splits on every predictor.
2. Pick the split with the most appropriate criteria.
3. Apply the split.
4. Iterate repeatedly for the two child nodes.

The algorithm stops when the node is pure, i.e. the impurity is zero. It is worth noting here that as far as classification is concerned, a node is pure if and only if it contains observations of only one class. We use the training dataset to train the model. The model is described by a decision tree. Splits are contained on the nodes within a tree in a way that a number of samples are distinguished based on gini impurity criterion.

2.5 Neural Networks

A huge parallel computing system with a large array of simple processes with many interconnections to it can be classically defined as neural networks. The obvious choice of the family of neural networks for pattern classification task is the feed-forward network, which includes multilayer perceptron. Such network consists of various layers with only unidirectional connections permitted between them.

Let us denote m input values by $\{x_1, x_2, \dots, x_m\}$. Each of the m inputs has a weight associated with them $\{w_1, w_2, \dots, w_m\}$. The input values are multiplied by their weights and summed

$$v = w_1x_1 + w_2x_2 + \cdots + w_mx_m = \sum_{i=1}^{i=m} w_ix_i \tag{17}$$

The output is some function $net = f(v)$. This function is called the activation function of the network. The activation function gives non-linearity to the network. Sigmoidal function can be used as the activation function of the network. Sigmoidal function has many desirable properties which we want for an activation function. It is non-linear (exponential) function with range$(0,1)$. The usage of a number of layers, along with a non-linear activation function makes neural networks universal function approximators.

3. SIMULATION AND TESTING

This section describes the results obtained from different classification algorithms when applied on process data. All the tests are done in a Matlab environment. The dataset is taken from a real process plant. All data are numeric. The size of dataset is 3719×37. Linear and Logistic Regression models are trained using the methodology described earlier. For training of CART model, built-in function *fitctree* is used; while for Neural Networks (NN), *patternnet* is used for creation of NN model and is used to train the model.

Fig. 1 below represents the data samples from training set. For the purpose of visualization, three attributes are selected using the idea described later in this paper. The samples are colored according to the predefined category that each samples belong to. The sample represented in blue signifies the normal operating conditions; green and red represents abnormal operating conditions.

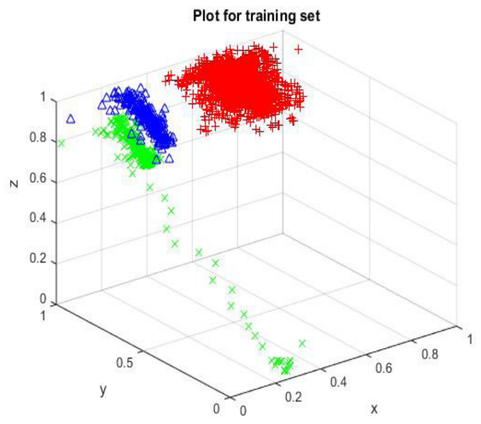

Figure 1: Plot for training set

Fig. 2 below represents the plot of the samples of the test set along with the result of the classifier on the test set (shown in different colors as described earlier). It can be observed that the classifiers are able to predict the class of samples accurately for most of the cases.

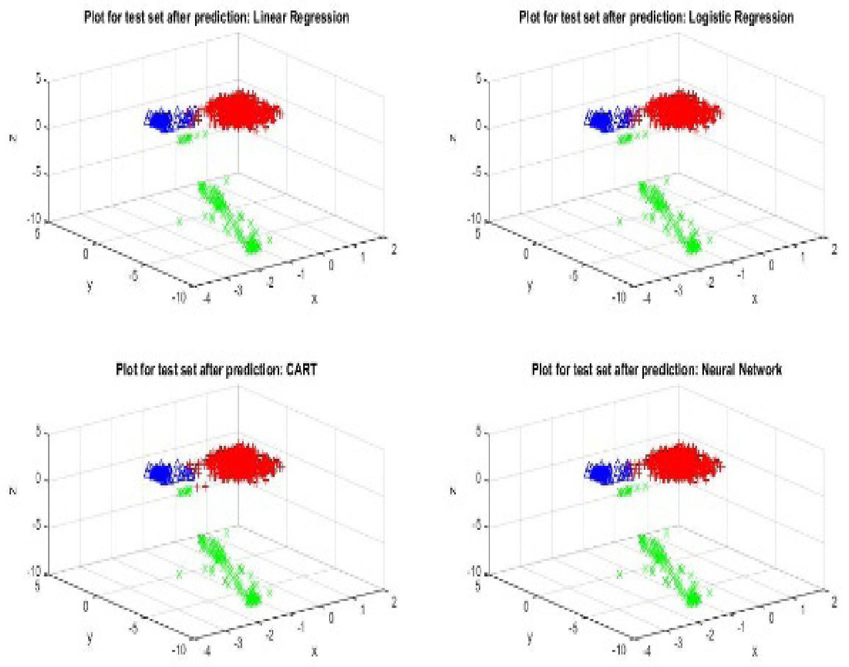

Figure 2: Plot representing the samples of test set after prediction

Fig. 3 below shows the decision tree obtained as a result of the CART model. It can be observed that the decision tree has only four nodes i.e. only four attributes are selected to classify the input patterns into one of the categories, with category 0 representing normal values; 1 representing faulty values due to on type of fault incident; and 2 represents faulty values due to another fault incident.

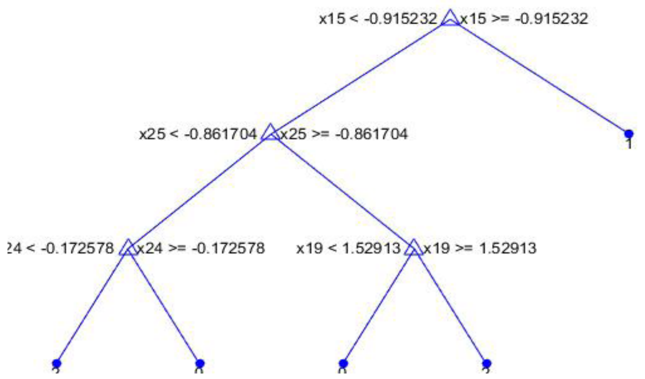

Figure 3: Decision tree obtained from CART model

4. CONCLUSION

It can be concluded from the results that data driven models are very useful in fault detection in process industries. The type and location of fault can also be studied by observing the values of the weights obtained in linear and logistic regression; also the attributes used as nodes in case of CART can also be used for indicating the location of fault.

The methods mentioned in the table given below have been applied using the datasets iteratively and thereby the accuracies achieved as given in table. The prediction methods used here has resulted in highly accurate models. Of all the methods used for training, CART gives the highest accuracy. But the accuracy of other algorithms are not far away from CART. Table 1 shows the accuracy of each algorithm.

Table 1
Accuracy of different classification algorithms

Methods	Accuracy (%)
Linear Regression	99.24
Logistic Regression	99.03
Classification and Regression Trees (CART)	99.57
Neural Networks	99.35

References

[1] K. Jain, M.N. Murty and P.J. Flynn, "Data Clustering: A Review", *ACM Computing Surveys,* Vol. 31, No. 3, September 1999.

[2] Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques", 2006.

[3] Richard O. Duda, Peter Hart, David Stork, "Pattern Classification", 2006.

[4] Trevor Hastie, Robert Tibshirani and Jerome Friedman, "The Elements of Statistical Learning: Data Mining, Inference, Prediction", 2008.

[5] Xindong Wu et. al., "Top Ten Algorithms in Data Mining", *Knowledge and Information Systems*, 2008.

[6] Wei-Yin Loh, "Classification and Regression Tree Methods", Encyclopedia of Statistics in Quality and Reliability, pp. 315-323, Wiley, 2008.

[7] Ryan Rifkin and Aldebaro Klautau, “In Defense of One-vs-All Classification”, *Journal of Machine Learning Research* 5, 2004, pp. 101-141.

[8] Golriz Amooee, Behrouz Minaei-Bidgoli and Malihe Bagheri-Dehnavi, “A Comparison Between Data Mining Prediction Algorithms for Fault Detection”, International Journal of Computer Science Issues, Vol. 8, Issue 6, No. 3, November 2011.

[9] Poonam Chaudhary & Vikram Singh, “Network Fault Detection—A Case For Data Mining”, *International Journal of Computing and Business Research (IJCBR),* Volume 5 Issue 4, July 2014.

[10] Reza Entezari-Maleki, Arash Rezaei, and Behrouz Minaei-Bidgoli, “Comparison of classification methods based on the type of attributes and sample size”, *Journal of Convergence Information Technology* 4 (3), 94-102, 2009.

[11] Byungchul Park, Young J. Won, Hwanjo Yu, James Won-Ki Hong, Hong-Sun Noh, and Jang Jin Lee, “Fault Detection in IP-based Process Control Networks using Data Mining”, *IEEE International Symposium on Integrated Network Management*, 2009.

[12] A. Khoukhi, H. Khalid, R. Doraiswami, L. Cheded, “Fault Detection and Classification using Kalman Filter and Hybrid Neuro-Fuzzy Systems”, *International Journal of Computer Applications* (0975 – 8887) Volume 45– No.22, May 2012.

[13] Fellipe do Prado Arruda, Anderson da Silva Soares, “Fault Detection in Industrial Plant Using K-Nearest Neighbors with Random Subspace Method”.

worldcomp-proceedings.com/proc/p2014/ICA3200.pdf

[14] Sathyan Munirathinam and Balakrishnan Ramadoss, “Predictive Models for Equipment Fault Detection in the Semiconductor Manufacturing Process”, *International Journal of Engineering and Technology,* Vol. 8, No. 4, April 2014.

[15] A.A. Shahrjooi Haghighi, M. Abbasi Dezfuli, S.M. Fakhrahmad,” Applying Mining Schemes to Software Fault Prediction: A Proposed Approach Aimed at Test Cost Reduction”, Proceedings of the World Congress on Engineering 2012 Vol I, July 4-6, 2012.

[16] Yair Meidan, Boaz Lerner, Gad Rabinowitz, and Michael Hassoun, “Cycle-Time Key Factor Identification and Prediction in Semiconductor Manufacturing Using Machine Learning and Data Mining”, *IEEE Transactions On Semiconductor Manufacturing*, Vol. 24, No. 2, May 2011.

[17] Davi L. de Souza, Matheus H. Granzotto, Gustavo M. de Almeida, Luís C. Oliveira-Lopes, “Fault Detection and Diagnosis Using Support Vector Machines - A SVC and SVR Comparison”, *Journal of Safety Engineering*, 3(1): 18-29, 2014.

[18] Binoy B. Nair, Vamsi Preetam M. T, Vandana R. Panicker, Grishma Kumar V and Tharanya A, ” A Novel Feature Selection method for Fault Detection and Diagnosis of Control Valves”, *IJCSI International Journal of Computer Science* Issues, Vol. 8, Issue 3, No. 1, May 2011.

[19] Elyas Rakshani, Imran Sariri, kumars Rouzbehi, Application of datamining on fault detection and prediction in boiler of power plant using artificial neural network”, *IEEE*, 2009.

[20] Nour El Islem Karabadji, Hassina Seridi, Ilyes Khelf, and Lakhdar Laouar, “Decision Tree Selection in an Industrial Machine Fault Diagnostics”, Springer, pp. 129–140, 2012.

[21] Kittisak Kerdprasop and Nittaya Kerdprasop, “A Data Mining Approach to Automate Fault Detection Model Development in the Semiconductor Manufacturing Process”, *International Journal of Mechanics*, Issue 4, Volume 5, 2011.

[22] Libo Bie, Xiangdong Wang, “Fault Detection and Diagnosis of Continuous Process Based on Multiblock Principal Component Analysis”, *International Conference on Computer Engineering and Technology*, 2009.

[23] Hossam A.Gabbar, Akinlade Damilola, Hanaa E. Sayed, “Trend Analysis Using Real Time Fault Simulation for Improved Fault Diagnosis”, *IEEE*, 2007.

[24] Yogendra Kumar Jain, Upendra, “Intrusion Detection using Supervised Learning with Feature Set Reduction International Journal of Computer Applications”, Volume 33–No.6, November 2011.

[25] Kai-Ying Chen, Long-Sheng Chen, Mu-Chen Chen, Chia-Lung Lee, “Using SVM based method for equipment fault detection in a thermal power plant”, Elsevier, 2010.

[26] Ezat mahmoud soleiman, Abdelhamid fetanat, “Intrusion Detection System Using Supervised Learning Vector Quantization”, *International Journal of Innovative Research in Advanced Engineering*, Volume 1 Issue 10, November 2014.

[27] Neeraj Prakash Srivastava, R. K. Srivastava, P. K. Vashishtha, ” Fault Detection and Isolation (Fdi) Via Neural Networks”, *Int. Journal of Engineering Research and Applications*, Vol. 4, Issue 1 (Version 1), January 2014.

[28] Tarun Chopra, Jayashri Vajpai, “Classification of Faults in DAMADICS Benchmark Process Control System Using Self Organizing Maps”, *International Journal of Soft Computing and Engineering (IJSCE)*ISSN: 2231-2307, Volume-1, Issue-3, July 2011.