2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Gas Identi cation by Using a Cluster-k-Nearest-Neighbor

Samir Brahim Belhaouari [1 +], Riadh Abaza [2]

[1]Tel:00 60 17 519 0032, fax: 006053655905, email:brahim belhaouari@petronas.com.my
[2]email:riadh.abaza@a3.epfl.ch

Abstract. Among the most serious limitations facing the success of future consumer gas identification systems are the drift problem and the real-time detection due to the slow response of most of todays gas sensors. In this paper, a novel gas identification approach based on Cluster-k-Nearest Neighbor. The effectiveness of this approach has been suc-cessfully demonstrated on an experimentally obtained data set. Our classify takes advantage of both k-NN which is highly accurate and K-means cluster which is able to reduce the time of classification, we introduce Cluster-k-Nearest Neighbor as "variable k"-NN dealing with the centroid or mean point of all subclasses generated by clustering algo-rithm. In general the algorithm of K-means cluster is not stable in term of accuracy. Therefore for that reason we develop another algorithm for clustering space which contributes a higher accuracy compares to K-means cluster with less subclass number, higher stability and bounded time of classification with respect to the variable data size. We find 98.7% of accuracy in the classification of 6 different types of Gas by using K-means cluster algorithm and we find almost the same by using the new clustering algorithm.

Keywords: Pattern recognition, Gas Sensor, k-Nearest Neighbor, k-means cluster, Gaussian Mixture Model, Classification

1. Introduction

GAS identification on a real-time basis is very critical for a very wide range of applications in the civil and military environments. The past decade has seen a significant increase in the application of multisensor arrays to gas classification and quantification. Most of this work has been focused on systems using microelectronic gas sensors featuring small size and low-cost fabrication, making them attractive for consumer applications. A number of interesting applications have also emerged in the last decade, whether related to hazard detection, poisonous and dangerous gases or to quality and environmental applications such as air quality control. A gas sensor array permits to improve the selectivity of the single gas sensor and shows the ability to classify different odors. In fact, an array of different gas sensors is used to generate a unique signature for each odor. After a preprocessing stage, the resulting feature vector is used to solve a given classification problem, which consists of identifying an unknown sample as one from a set of previously learned gases. Significant work has been devoted to design a successful pattern analysis system for machine olfaction [4].

Various kinds of flexible pattern recognition algorithms have been used for classifying chemical sensor data. Most notably, neural networks have been exploited, in particular multilayer perceptrons (MLPs) and radial basis functions (RBFs). Other methods based on the class-conditional density estimation have been used, such as quadratic and K nearest neighbor (KNN) classifiers. These parametric and nonparametric density estimation methods have their merits and limitations.

The most perfect classification would verify at least the three conditions below:

- Highly accurate,

+ Corresponding author. Tel.: +00 60 17 519 0032
E-mail address: brahim belhaouari@petronas.com.my.

- Minimum classification time,
- Smaller size of training data.

In our paper, we will take advantage of k-Nearest Neighbor, k-means cluster and Gaussian Mixture Model to make a good classifier which is fast and accurate .

Classification using Gaussian Mixture Model (GMM)[22] or k-Nearest Neighbor (k-NN) [10] are almost the same in the sense that they consider the neighbor data of the new pixel vector x, which will be classified and will be more accurate compared to NN. This is because they are more efficient in the overlapping area as these methods take more consideration to samples of training data that are less frequent.

To reduce the classification time for k-NN, we need to cluster our space (training data) into subclasses, where each subclass will be represented by one data (we can choose two or more representatives according to the number of subclasses), then we use classification algorithm of NN (or k-NN) to classify by using representative data. Each subclass contains a random number of data, which are relatively close to each other. We call this manner of classification as Cluster-k-NN (C-k-NN) which is similar to 'variable k'-NN.

The classification time in NN and k-NN are of the order N^2, i.e., O(N^2), where N is the training data size. To reduce the time we need to cluster our space, in fact the time of classification will depend just on the number of subclasses m_i, with m_i the number of subclasses in the class C_i, thus the C-k-NN algorithm. In general m_i is a small number and does not depend on the training data size (the number of Gaussian functions to estimate a probability density, for GMM method, is in general bounded with respect to the variable N).

The most widely used method of non-parametric density estimation is k-NN [1]. k-NN rule is a powerful technique that can be used to generate highly nonlinear classification with limited data. To classify a pattern x, we find the closest k examples in the dataset and select the predominant class C_i among those k neighbors (problem if two or more classes are predominant class!). One drawback of k-NN is that the training data must be stored, and a large amount of processing power is needed to evaluate the density for a new input pattern. However C-k-NN correct those drawbacks points.

The k-NN classifier is generally based on the Euclidean distance between a test sample x and the specified training samples but in C-k-NN we introduce other metrics in order to better estimate the density probability. Let $x_{i,j}$ belongs to the subclass j of the class i, we note $C_{i,j}$, and for all positive number s we can define the following metric:

$$d(x,x_{i,j}) = \frac{d^S_{euclidean}\left(x, \hat{x}_{i,j}\right)}{n_{i,j}}, \forall x \in R^d$$

where $x_{i,j}$ is the representant of $C_{i,j}$ and $n_{i,j} = card\left(C_{i,j}\right)$ or it can be the variance of the set $C_{i,j}$.For parametric method we have Gaussian Mixture Model [22] which is classified as a semi-parametric density estimation method since it defines a very general class of functional forms for the density model. In a mixture model, a probability density function is expressed as a linear combination of basis functions. A model with M components is described as mixture distribution

$$P(x) = \sum_{j=1}^{M} P(j) P(x/j)$$

where P(j) are the mixing coefficients and P(x/ j) are the component density functions. Each mixture components is defined by a Gaussian parametric distribution in d dimensional space

$$P(x) = \sum_{j=1}^{M} P(j) P(x/j) = \frac{\exp\left\{-\frac{1}{2}(x-\mu_j)^T \sum_j^{-1}(x-\mu_j)\right\}}{(2\pi)^{d/2}\left|\sum_j\right|^{1/2}}$$

The parameter to be estimated are the mixing coefficients P(j), the covariance Matrix $\sum_j$ and the mean vector μ_j,

In C-k-NN we approximate each Gaussian function, P(j)P(x= j), by $\frac{1}{cst + d\left(x - \mu_j\right)}$, where cst is any small number, we add it just to avoid the division by 0. To estimate the number of subclasses and their representatives for C-k-NN(or the number of Gaussian functions, M, and their means μ_j for GMM) we can use k-means cluster, or another new similar stable algorithm which will be explained later. The k-means cluster algorithm or the modified one needs the number of subclasses as input, and to fixed the number of clusters we iterate the number of clusters starting from one and we take the following conditions as stopping criterium:

Table 1: Classification results comparisons

	Cluster-k-NN	Cluster-k-NN*	NN
Accuracy	98.7%	98.2%	96%
Subclass number	[11,5,24,20,33,21]	[5,6,7,7,55,44]	[50,25,50,106,111,90]
Classification time	26.3% of T	28% of T	T=*O(datasize)*

*for modified algorithm of k means cluster with α = 0:085 and α' = 0:82.

a. All the representatives or centroid $\left(\mu_{i,j}\right)$ have to be closer to their class C_i (or $C_{i,j}$) than to other classes. This is to decrease the misclassification (i.e. no error in the classification of our own training data)

b. the variance of each class, var, does not decrease considerably in comparison to the previous iteration. We define the variance of each class as var=$\sum_{i=1}^{n}$ var$_i$ where var$_i$ is the variance of the subclass i and we can take as criterium of smooth function var with the number of subclasses as variable. In our simulation α = 0:9 gives the best result.

$$\frac{\Delta \text{var}}{\text{var}} \leq \alpha$$

Our dictionary is well defined now, for each class C_i is represented by $\left\{\mu_{i,1}, \mu_{i,2}, ..., \mu_{i,mi}\right\}$, $1 \leq i \leq cn$ where m_i is number of subclasses for the class C_i and cn is number of classes. To classify a new pattern x we use NN algorithm or k-NN algorithm (with small k) on the dataset $\left\{\mu_{i,j} : 1 \leq i \leq cn, 1 \leq j \leq m_i\right\}$, which means we consider the minimum distance rule, and assign x to the class C_i which is verified

$$C_i = \arg\left(\min_{\substack{1 \leq i \leq cn \\ 1 \leq j \leq mj}} \left\{d\left(x, \mu_{i,j}\right)\right\}\right)$$

where arg $\left(d\left(x, \mu_{i.j}\right)\right) = C_i$,for all $1 \leq cn$ The algorithm of k-means cluster is not stable since the result depends on the random choice of k initial vectors.

Therefore we develop another way to initialize this algorithm, which in general gives a better result than the classic k-means cluster e.i.,

$$Var_{\text{modifiedalgorithm}} \leq Var_{\text{classicalgorithm}}$$

For validation we use real data coming from gas sensor, we select six gases CH, CH4CO, CO, EthaCH, EthaCO and Ethanol. This dataset contains 25, 106, 50, 90, 111 and 50 examples of gas sensor response for each gas, The characteristics of the datasets are summarized in Table 2 We will explain more about data description in Section 3.

The test result shows that the proposed approach give a good accuracy with bounded time of classification, independently with the size of data, as shown in the Table 1

The reminder of this paper is organized as follows: Section (2) focused on our proposed method where two new algorithms are introduced, both of them are helpful to initialize the k-means cluster algorithm i.e., the choice of initial vectors. Section (3) emphasizes on the explanation of experiments and results. Finally section (4) summarizes the presented research work.

2. Method

Each class, C_i, should be a cluster to several subclasses, $C_{i,j}$, with $1 \le j \le m_i$, and each subclass will be represented by it means, $\mu_{i,j}$. Thus the cluster analysis seeks to identify a set of groups, which minimize the within-group variation and maximize the between-group variation.

We apply k-means cluster algorithm for each class in order to do the clustering, then we need to define the number of subclasses for each class and the initial k-vectors to initialize the k-means cluster algorithm.

To find the best suitable number of subclasses, we iterate the number of subclasses starting from 1 with two conditions to stop the iteration:

- All the representative, $\mu_{i,j}$, should be close,in respect to the metric d, to their class C_i i.e., if we classify all the representatives $\mu_{i,j}$ we have to find 100% of accuracy.If there are some misclassifications of $\mu_{i,j}$, we have to decrease the parameter α by multiply it by a another factor, α', less than 1.
- The variance of each class C_i, var_i, does not decrease considerably in comparison to the previous iteration.We can use $\frac{\Delta \text{var}}{\text{var}} \le \alpha$ as a criterium to quantify if there is a decrease or if it is approximately still constant. In certain case, it is better to stop the iteration if the condition, $\frac{\Delta \text{var}}{\text{var}} \le \alpha$, has been checked twice or more, i.e.,after where the variance will be smooth.

For initialization of k-means cluster algorithm in general we choose aleatory k-vectors, which belong to our class data. This makes the algorithm unstable in the sense of the final variance, depending on the initial vectors.

$$\text{var}\, c_i \sum_{j=1}^{m_i} \text{var}\, c_{i,j}$$

From here, the question ”How to choose the initial vectors in order to find a minimal variance?” arises. In this paper we develop two algorithms: near-to-near and near-to-mean, which we may do some modification related to each application.

2.1. Near-to-Near algorithm

This algorithm consists of calculating the distance $d\left(x_{i,n}, x_{i,m}\right)$ for all $x_i \in C_i$ and starting to cluster our class to $N_i - 1$ subclasses, where $coad\left(C_i\right) = N_i$. We put the two closest data at the same subclass $C_{i,1} = \left\{x_{i,n0}, x_{i,m0}\right\}$, where

$$\min_{n \neq m} d\left(x_{i,n}, x_{i,m}\right) = d\left(x_{i,n0}, x_{i,m0}\right)$$

and we put each other data at separate subclass,

$$\mathrm{C_{i,j}} = \left\{x_{i,j}\right\}, \forall j \in \{1,...,N\} - \{n_0, m_0\}$$

The next step the following index n_1 and m_1 are considered for which

$$\min_{\substack{n \neq m \\ (n,m) \neq (n_0, m_0)}} d\left(x_{i,n}, x_{i,m}\right) = d\left(x_{i,n1}, x_{i,m1}\right)$$

If $x_{i,n1}$ and $x_{i,m1}$ belong to the same subclass $C_{i,r}$,we split this subclass in to two other subclasses

$$C_{i,r+1} = C_{i,r} - \left\{x_{i,n1}, x_{i,m1}\right\} \quad (1)$$

$$C_{i,r} = \left\{x_{i,n1}, x_{i,m1}\right\} \quad (2)$$

But in the case if $x_{i,n1}$ and $x_{i,m1}$ belong to two di_erent subclasses $C_{i,r1}$ and $C_{i,r2}$ respectively, we put $x_{i,n1}$ in the subclass $C_{i,r2}$ if $coad\left(C_{i,r2}\right) > coad\left(C_{i,r1}\right)$ and we put $x_{i,m1}$ in the subclass $C_{i,r1}$ if

$coad\left(C_{i,r2}\right)\_coad\left(C_{i,r1}\right)$. Indeed to use the cardinally of set we can use the distance between the vector to the set as d(vector; mean of set).

When we obtain k subclasses, we stop the iteration, and our initial k-vectors will be the mean of each subclasses.

2.2. Near-to-Mean algorithm

This algorithm is almost the same as the Near-to-Near one, but we will deal with the mean of subclass $C_{i,r}$ in the processing. We start to split our class C_i into two subclasses

$$C_{i,1} = \left(x_{i,n0}, x_{i,m0}\right)$$

and

$$C_{i,2} = \left\{x_{i,j} \,\middle|\, j \notin \left\{n_0, m_0\right\}\right\}$$

where $d\left(x_{i,n0}, x_{i,m0}\right) = \min_{n \neq m}\left(x_{i,n}, x_{i,m}\right)$

We update our class C_i by replacing $x_{i,n0}$ and $x_{i,m0}$ by their average,i.e.

$$C_i^1 = \left\{..., x_{i,n0-1}, s_0, x_{i,n0+1}, ..., x_{i,m0-1}, s_0, x_{i,m0+1}, ...\right\}$$

where $s_0 = \left(x_{i,n0} + x_{i,m0}\right)/2$

Next we consider $x_{i,n1}$ and $x_{i,m1}$ such as

$$d\left(x_{i,n1}, x_{i,m1}\right) = \min\left\{d\left(x_{i,n}, x_{i,m}\right) \middle| d\left(x_{i,n}, x_{i,m}\right) \neq 0\right\}$$

We replace all the data in C_i^1 that are equal to $x_{i,n}$ or $x_{i,m}$ by s_1, which is the mean of the union of the two subclasses where $x_{i,n1}$ and $x_{i,m1}$ belong to

$$s_1 = \frac{C_{n1}x_{i,n1} + C_{m1}x_{i,m1}}{C_{n1} + C_{m1}}$$

where C_{n1} is the number of repetition of $x_{i,n1}$ inside of C_1 and C_{m1} is the number of repetition of xi;m1 inside of C_1.

Our algorithm stops once the number of distinct vectors inside of C_i^r is equal to k.

Our algorithm of classification does not need to keep all the data, instead it only need the average of each subclass.

This is one of the powerful point of the clustering. To classify a new data or vector x, we use k-NN algorithm,i.e., we assign x to the class $C_{\tilde{i}}$ for which

$$\tilde{i} = \arg_i \min_{i,j} d\left(x, \mu_{i,j}\right)$$

where $\arg_i d\left(x, \mu_{i_0,j_0}\right) = i_0$.

Further investigation about the k-NN algorithm is indeed to find the closest j- examples in the dataset and select the predominant class. We can find the smallest closest examples in the dataset and select the predominant class which have exactly k examples.

3. Experiments and Results

We evaluated the performance of the proposed Cluster-k-NN classifier on six datasets of gases collected from both a commercial Taguchi gas sensors (TGS) and a microhotplate (MHP) microelectronic gas sensors arrays.

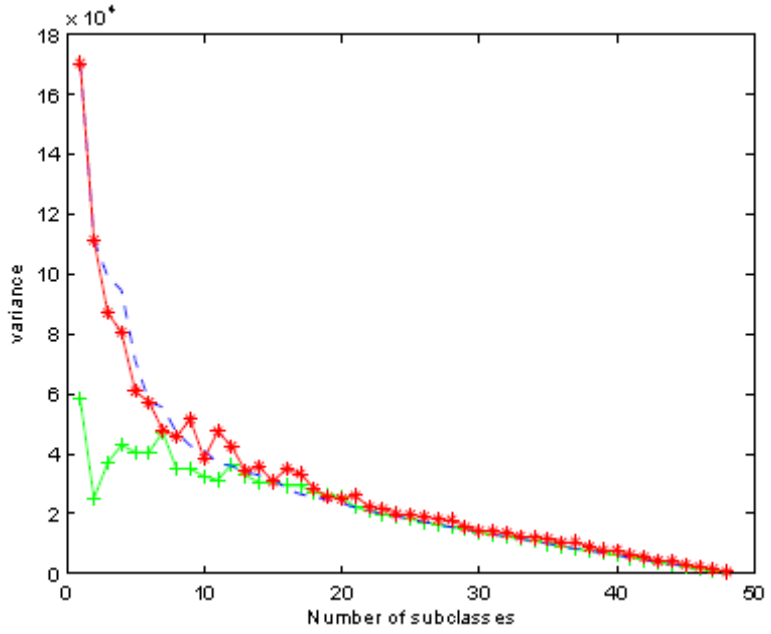

Figure.1: Graphs of variance in function of number of subclasses by using k-means cluster, the ”+” with Near To Near algorithm ,”-” with Near To Mean algorithm and ”*” with aleatory initialization.

Table 2: DATA SET CHARACTERISTICS

Data set	Type of gas	Number of patters	Number of sensors
1	CH	25	8
2	CH4	106	8
3	CO	50	8
4	EthaCH	90	8
5	EthaCO	111	8
6	Ethanol	50	8

3.1. Data Description

Measurements have been done with an experimental equipment consisting of gas pumps, mass flow controllers, a sensor chamber, and a computer used for data acquisition and the experiment control. In a gas chamber, we placed a sensor array based either on the commercial TGS or MHP microelectronic gas sensors. Vapors were injected into the gas chamber at a flow rate determined by the mass flow controllers. Measurement procedure consists of two steps.

The first step consists of injecting the tested gas during the 10-min period, whereas 40 min is allocated to a cleaning stage with dry air. Data are collected at a sampling period of 3 s during 3*16 s. The necessity of fast classification algorithm come from the shortest of sampling interval of the sensor response.

By using Matlab software, we simulate our algorithms. The test results shown in Table 2, indicates us that the new approach gives a remarkable accuracy with limited time of classification, independency with the size of the training data N. As a conclusion k-NN has the advantage of not needing to estimate the density function. However, its high computational load makes it unsuitable for hyperspatial data classification. Using the idea of k-means cluster we can remove the time drawback.

Table 3: CLASSIFICATION ACCURACY (%) WITH SEARCH INTERVAL

Dimension	4	6	8	12	16
Accuracy	98.7%	98.7%	98.2%	98.7%	98.7%

The Table 3 shown that the search interval [0; 12s] is enough to do the classification at 98.7% of accuracy,i.e., for each pattern we collect 4 points, at 3s, 6s, 9s and 12s. This result is suitable for real time applications.

Figure 1 shows that if we choose valid initial vectors for k-means cluster we can obtain a small variance. In this case, the variance of Near-to-Near algorithm is smaller than the variance of Near-to-Mean algorithm if the number of subclasses is bigger than 5. The Near-to-Mean algorithm deals better if the number of subclasses is small than the Near-to-Near algorithm

4. Conclusion

Classification by using Cluster-k-Nearest Neighbor with Near-To-Near algorithm is an almost perfect method of classification in terms of accuracy and bounded time of classification through taking advantage from K-Nearest.Neighbor for it accuracy, and from Gaussian Mixture Model and clustering for their reducing to classification time.Near-to-Near algorithm gives a reasonable set of vectors to initialize K-means cluster algorithm in order to minimize the variance of space data. We have developed another algorithm for clustering with a global minimum of variance, but we still trying to reduce time of calculation. We suspect such that algorithm can improve our classifier.

5. References

[1] B. V. Dasarsthy, Ed., Nearest Neighbor (NN) Norms: NN Pattern classification techniques. Los Alamitos, CA:*IEEE Computer Society Press,*1990.

[2] P. J. Hardin, Parametric and Nearest Neighbor methods for hybrid classification: A comparaison of pixel assignment accuracy, photogramm *Eng. Remote Sens.,* Vol. 60 pp. 1439-1448, Dec. 1994.

[3] T. P. Yunck, A technique to identify NN, *IEEE Trans. Syst., Man, Cybern,*. Vol. SMC-6, no. 10, pp. 678-683, 1976.

[4] S. Brahim Belhouari and A. Bermak, Gaussian process for nonstationary time series prediction, *Computat. Stat. Data Anal.*, vol. 47, no. 4,pp. 705712, Nov. 2004.

[5] S. Brahim-Belhouari, A. Bermak, G. Wei, and P. C. H. Chan, in *IEEE TENCON*, Vol. A, p. 693, Chiang Mai, Thailand, 2004.

[6] Amine Bermak1, Sofiane Brahim Belhouari1, Minghua Shi1, *Dominique Martinez2 Pattern Recognition Techniques for Odor Discrimination in Gas Sensor Array*, www.aspbs.com/eos.

[7] Sofiane Brahim-Belhouari, Amine Bermak Fast and Robust Gas Identification System Using an Integrated Gas Sensor Technology andGaussian Mixture Models, *IEEE SENSORS JOURNAL*, VOL. 5, NO. 6, DECEMBER 2005.

[8] Amine Bermak and Sofiane Brahim Belhouari Bayesian Learning Using Gaussian Process for Gas Identification, *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, VOL. 55, NO. 3, JUNE 2006.

[9] X. Jia and J. A. Richards, Cluster-space representation for hyperspectral data classification, *IEEE Trans, Geosci. Remote Sens.*, vol. 40, no.3, pp. 593-598, Mar. 2002.

[10] X. Jia and J. A. richards, Fast k-NN classification Using the Cluster-Space Approach, *IEEE Trans., Geosci. Remote Sens.*, vol. 2, no. 2, April 2005.

[11] K. wagsta_ and S. Rogers, Constrained k-means Clustering with Background Knowledge, *Proc. of the 8th International conference on Machine learning*, p. 577-584, 2001

[12] Pentakalos, O., Menasce, D., and Yesha, Y., Automated clustering-based workload characterization, Joint Sixth NASA Goddard Space Flight Center Conference on Mass Storage Systems and Technologies and Fifth IEEE Symposium on Mass Storage Systems, *College Park*, MD,March 23-26, 1998.

[13] Jain, A., Murty, M., Flynn, P., *Data Clustering*: A Review, http://scgwiki.iam.unibe.ch

[14] P. C.H. Chan, G. Yan, L. Sheng, R. K. Sharma, Z. Tang, J.K.O. Sin, I.-M. Hsing, and Y. Wang, An integrated gas sensor technology using surface micro-machining, Sens. *Actuators B*, vol. 82, pp. 277283, 2002.

[15] J. W. Gardner and P. N. Barelett, A brief history of electronic noses, Sens. *Actuators B*, Chem., vol. 18/19, no. 13, pp. 211220, Mar. 1994,

[16] Elsevier Sequoia.

[17] Eisen, Michael. Cluster 3.0 Manual. http://bonsai.ims.u-tokyo.ac.jp

[18] Hale. Introductory Statistics, http://espse.ed.psu.edu/statistics.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2000.

[20] I. T. Nabney, Netlab, *Algorithms for Pattern Recognition*. London, U.K.: Springer-Verlag, 2003.

[21] C. K. I. Williams and D. Barber, Bayesian classification with Gaussian processes, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12,pp. 13421351, Dec. 1998.

[22] D. J. C. MacKay. (1997). Gaussian processesA replacement for supervised neural networks, in Lecture Notes TutorialNIPS . [Online]Available: http://www.inference.phy.cam.ac.uk/mackay/gp.pdf.

[23] D. M. Titterington, A. F. M. Smith, and U. E. Makov, Statistical Analysis of Finite Mixture Distributions. *John Wiley*, New York, 1985.

[24] C. K. I.Williams and C. E. Rasmussen, Gaussian process for regression, in Advances in Information Processing Systems, 8th ed. *Cambridge*,MA: MIT Press, 1996, pp. 111116.

[25] C. M. Bishop, Neural Networks for Pattern Recognition, *Clarendon Press, Oxford*, 1995.

[26] S. Brahim-Belhouari and A. Bermak, Pattern Recognition Letters, *Elsevier Science Publisher*, 2005.

[27] S. Brahim-Belhouari, Modified k-means cluster. *University technology of PETRONAS*, 2008.

[28] Jenn-Hwai Yang, Miin-Shen Yang,A control chart pattern recognition system using a statistical correlation coe_cient method, *Elsevier*,*Computers and Industrial Engineering* 48 (2005) 205221.

[29] Dan Li1, Sang C. Suh2, Jingmiao Gao3, A NEW TIME-SERIES CHART PATTERN RECOGNITION APPROACH, *Integrated Design and Process Technology, IDPT-2005 @2005 Society for Design and Process Science.*

[30] Seref SAGIROGLU,Erkan BESDOK, Mehmet ERLER, *Control Chart Pattern Recognition Using Artificial Neural Networks*,Turk J Elec

[31] Engin, VOL.8, NO.2 2000,c TUBI-TAK.