Proceedings of International Workshop on Data Mining and Artificial Intelligence (DMAI' 08)
24 December, 2008, Khulna, Bangladesh

Detection of faulty products using data mining

M. A. Karim[a*], G. Russ[b], and A. Islam[c]
[a]*School of Engineering System*
Queensland University of technology, QLD 4001
[b]*Faculty of Informatics*
University of Magdeburg, Germany
[c]*Build and Integration Engineering, Nokia Devices*
Hoimotie 19, 00380 Helsinki, Finland

Abstract

The manufacturing process is complex due to the large number of processes, diverse equipment set and nonlinear process flows. Manufacturers constantly face yield and quality problems as they constantly redesign their processes for the rapid introduction of new products and adoption of new process technologies. Solving product yield and quality problems in a manufacturing process is becoming increasingly difficult. There are various types of failures and their causes have complex multi-factor interrelationships. High innovation speed forced today's manufacturers to find failure causes quickly by examining the historical manufacturing data. Data mining offers tools for quick discovery of relationships, patterns, and knowledge in large databases. This has been applied to many fields such as biological technology, financial analysis, medical information, etc. Application of data mining to manufacturing is relatively limited mainly because of complexity of manufacturing data. Growing self-organizing map (GSOM) algorithm has been proven to be an efficient algorithm to analyze unsupervised DNA data. However, it produced unsatisfactory clustering when used on some manufacturing data. Moreover, there was no benchmark to monitor improvement in clustering. In this study a method has been proposed to evaluate quality of the clusters produced by GSOM and to remove insignificant variables from the dataset. With the proposed modifications, significant improvement in unsupervised clustering was achieved with complex manufacturing data. Results show that the proposed method is able to effectively differentiate good and faulty products.

1. Introduction

Manufacturing databases usually comprise of process control, process step, and quality control data. In many applications data are automatically generated by sensors and therefore the number of inputs can be very large. This large volume of data coupled with quicker time to market expectations is making finding and resolving problems quickly an overwhelming task. The analysis of such a large volume of data, interpreting results, and implementing design improvements is computationally intensive and time-consuming. A high priority goal for today's manufacturing is finding the most probable causative factor(s) that discriminate between low yield and high yield products by quickly examining the historical manufacturing data.

Engineers often perform a series of experiments to identify and resolve potential problems with the design. Traditional techniques of dealing with quality problems are the use of statistical process control (SPC) and design of experiments (DOE). However, these techniques fail to extract underlying features from complex data [1]. Moreover, these methods are highly time-consuming. In order to decrease design cycles and the time-to-market new products, it is important to have a method for analyzing manufacturing data quickly and efficiently, predicting the effects of design changes and determining the best design parameters. Effective product data management has a significant influence on product quality improvement [2, 3]. However, traditional techniques

* Corresponding Author: (617) 3138 1516 , Email: azharul.karim@qut.edu.au

1-4244-2136-7/08/$20.00 ©2008 IEEE

are incapable of dealing with large volumes of manufacturing data [4].

Advances in the data mining techniques enhanced automatic knowledge discovery as well as extraction of useful information from large volumes of data. Literature shows that the dynamic GSOM is a powerful tool for unsupervised data analysis [5]. In this study, a modified GSOM is used to detect faulty products from manufacturing data. Several modifications are proposed in order to reduce noisy data and to improve clustering quality.

2. Unsupervised Clustering Method

Cluster analysis is an unsupervised data mining technique that processes data that do not have known class labels, or where the analyst opts not to use them. Data within a cluster are highly similar to one another, but are very dissimilar to data in other clusters. Each cluster can be treated as a class of data, on which class or concept characterisation and discrimination can be performed.

The neural network approach to data mining has gained popularity due to the facts that most real world data have complex decision boundaries and contain a substantial amount of noise and the neural networks are robust against noise. A popular and well-accepted self-organising method of neural network analysis is self-organising maps (SOMs) [6]. A SOM with two-dimensional constraints can provide a visualisation of high-dimensional data by projecting it into a two dimensional network. However, the two-dimensional SOMs are rectangular grids of neurons that have predefined sizes and are specified by widths and heights. This can potentially hinder the real representation of input space topology [7].

Consequently, several modifications to SOM have been proposed to overcome the problem of predefined grids. One of the most recent variants, called the growing self-organising map (GSOM) was proposed to let the algorithm determine the size of the feature map [7-9]. A significant difference between these two algorithms is that SOM is not able to grow but GSOM grows according to its own growing criterion. Hsu et al. [9], proposed a hierarchical clustering algorithm using multiple layers of GSOM that automatically suggests an appropriate number of clusters that are present in the data.

2.1. GSOM algorithm

GSOM is an unsupervised learning algorithm and the basic idea is to:

1. Represent high-dimensional data in a low-dimensional form without loosing any of the important features of the data.

2. Organise data on the basis of similarity by grouping objects that are geometrically close to each other.

This makes GSOM popular for visualizing high-dimensional data. Though computationally more expensive than other unsupervised methods, it has proved to be extremely useful with high-dimensional and complex input data.

A GSOM consists of neurons that are usually arranged in a two-dimensional grid with rectangular or hexagonal topology. Each neuron is represented by an n-dimensional weight vector $m_i = [m_{i1}, m_{i2}, ..., m_{in}]$, where, n is equal to the respective dimension of the input vectors. The neurons are connected to adjacent neurons by means of neighbourhood relations.

The GSOM utilizes a competitive learning method [9]. When a training sample is given to the network, its distance to all weight vectors is computed. The best-matching unit (BMU) or winning neuron is the unit whose weight vector possesses greatest similarity with respect to the input sample **x**. The distance measure used to define the similarity is typically Euclidean distance. Thus for each input vector **x**, there exists a winning neuron m_b with minimum Euclidian distance to input **x**, such that

$$\|x - m_b\| = \min_i \{\|x - m_i\|\} \qquad (1)$$

This is referred to as the competitive learning part of the learning process, since all neurons compete to be the winner. Having found the BMU, the weight vectors of the GSOM are updated. The weight vectors of the BMU and its topological neighbours are moved closer to the input vector from the input space. This phenomenon is illustrated in Figure 1. The magnitude of the change decreases with time and is smaller for neurons physically far away from the BMU.

The update rule for changing the respective weight vectors of unit i is:

$$m_i(t+1) = m_i(t) + \alpha(t) h_{bi}(t)[x(t) - m_i(t)] \qquad (2)$$

where, t denotes the time step, x(t) is the input vector chosen from the input data set at time t, $\alpha(t)$ is a monotonically decreasing learning rate, and $h_{bi}(t)$ the

neighbourhood kernel around the winner unit at time t. The neighbourhood kernel defines the region of influence that the input sample has on the GSOM. This is called cooperative learning.

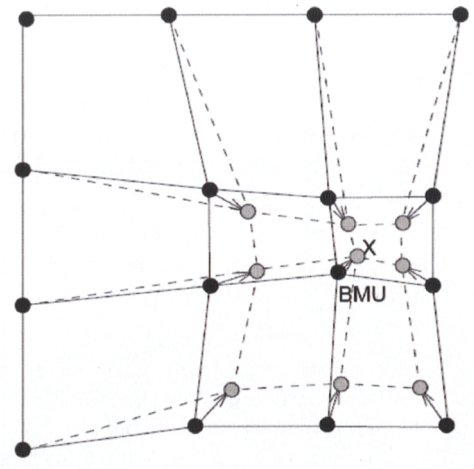

Figure 1: Updating the BMU and its neighbors towards the input sample [10]

There are many possible types of neighbourhood functions, the simplest and most commonly used are the following [10]:

- Bubble: In this simplest form it is one for all neurons close enough to BMU and zero for others
- Gaussian: $e^{-\frac{\|r_b - r_i\|^2}{2\sigma^2(t)}}$

where, $\mathbf{r}_b$ and $\mathbf{r}_i$ are positions of the winning neuron b and *i* respectively, and σ(t) is the neighborhood radius. The learning rule makes the weights of the winning neuron and its neighbours more similar to the input x. Regardless of the functional form, the neighbourhood function shrinks with time. For each input vector this process is repeated for large number of cycles. The final network associates the output nodes with groups or patterns in the input data set. During the training process a map is built and the neural network organises itself using a competitive process. As the weights of the whole neighborhood are moved in the same direction during the training phase, similar items tend to excite adjacent neurons. Therefore, GSOM forms a semantic map where similar samples are mapped close together and dissimilar apart. An analyst can then evaluate the map with the knowledge of input dataset.

The characteristic feature distinguishing neural maps from other neural network paradigms and regular vector quantisers is the preservation of neighbourhoods. This desirable feature obviously depends on the choice of the output topology. The proper dimensionality of the output space is usually not known a priori, but has to be specified prior to learning in the GSOM algorithm .

GSOM starts with a minimal number of nodes (usually four) and grows new nodes on the boundary based on a heuristic. By using the value called the spread factor (SF), the data analyst has the ability to control the growth of the GSOM. A parameter of growth known as, the growth threshold (GT) is defined as [9]:

$$GT = -D \times \ln(SF) \qquad (3)$$

where, D is the dimensionality of data and SF is the user defined spread factor that takes values between 0 and 1. An SF value close to 0 represents minimum growth and close to 1 represents maximum growth. Note that the outer boundaries of the spread factor (0 and 1) cannot be included since ln(0) = undefined, and ln(1) = 0 renders the growth threshold meaningless.

The algorithm first identifies the winning node. Then an accumulated error E (difference between input vector and weight) of the winning node is updated by the following rule:

$$E(t+1) = E(t) + |x - m_b| \qquad (4)$$

where, $\boldsymbol{x}$ is the input vector and m_b is the weight vector of the winning node. If the winning node is the boundary node and E exceeds GT, growing is initiated on that node to fill the surrounding unoccupied spaces of the lattice. If E of the winning node exceeds GT but the winning node is not a boundary node, then E is propagated outwards to other neighboring nodes. Weights of the new nodes will be initialized according to the following equation [3]:

$$m_{new} = 2m_{winner} - m_{opposite} \qquad (5)$$

where m_{opposite} represents the weight of the node topologically opposite to the new node. If there are no topologically opposite nodes, weights of the new nodes will be calculated according to the following equation:

$$m_{new} = m_{winner} + m_{other1} - m_{other2} \qquad (6)$$

where, m_{other1} and m_{other2} are weights of the nodes nearest to the new node. For hexagonal topology,

equation (6) is applicable as there will always be a neighbor of the winning node that is topologically on the opposite side of the new node. Growing phenomenon in GSOM is illustrated in Figure 2. New nodes are inserted around the high 'error node' (node 'Error' in Figure 2), with accumulated error exceeding the predefined growth threshold.

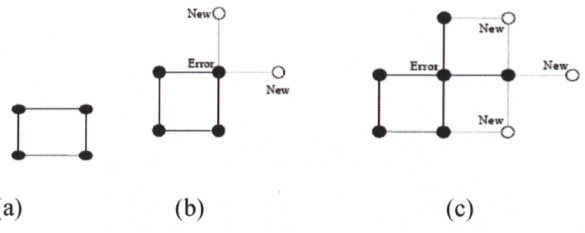

Figure 2: Growth in GSOM: (a) initial state (b) two new nodes and (c) three new nodes

2.2. Clustering quality improvement

Main objective of GSOM is to transform high-dimensional data in a low-dimensional cluster maps. However, with complex datasets, it may not be straightforward to obtain a good clustering. It is more applicable for manufacturing datasets, which are considered as complex due to the large number of processes, diverse equipment set, and nonlinear process flows. Manufacturing datasets often comprise of hundreds of process control, process step and quality control data. It might be necessary to change different variables and monitor the change of 'quality' of the cluster. This study proposes a clustering quality (CQ) measure, as a benchmark to evaluate the changes in clustering quality with different parameter changes. Mathematically, clustering quality can be expressed as:

$$CQ = \sum_i \max\left\{\left(\frac{\frac{g_i}{n_i} - \frac{G}{N}}{1-\frac{G}{N}} * \frac{n_i}{N}\right), 0\right\} + \sum_i \max\left\{\left(\frac{\frac{b_i}{n_i} - \frac{B}{N}}{1-\frac{B}{N}} * \frac{n_i}{N}\right), 0\right\} \quad (7)$$

where, B is total number of faulty products, G is total number of good products, N is total number products (B+G), b_i is number of faulty products in neuron i, g_i is number of good products in neuron i, and n_i = number of products in neuron i ($b_i + g_i$).

A CQ of 1 would mean a perfect separation of good and faulty products and a 0 would mean no separation at all. This proposed CQ takes the complete cluster map into account and can be generated automatically as an objective quantifier of the programme to separate good and faulty products from the data provided

Many datasets (manufacturing databases for example) are large and complex and may have substantial amounts of noise. Noise reduction from the data is one of the primary conditions for obtaining good clustering. Some datasets may contain many categorical variables, which are comprised of letters or a combination of numbers and letters. These categorical data should be transformed into numerical data to allow their use in data mining programmes. One of the ways to transform categorical data to numerical data is to use the binary convention.

Investigation of sample manufacturing datasets has revealed that there are many categorical variables that do not have any affect on separating good and faulty products. For example, a categorical variable may have an equal distribution among good and faulty products. These variables, when expanded, only add noise to the dataset. To filter out these unnecessary categorical variables, a method is proposed to remove them from the dataset. Mathematically the constraint can be expressed as:

$$FI = \left(\frac{X_{ig}}{N_g} - \frac{X_{if}}{N_f}\right) \le \alpha \quad (8)$$

where, FI = Filtration index, X_{ig} = number of good products having a particular categorical variable , X_{if} = number of faulty products having a particular categorical variable, N_g = number of good products in the sample, N_f = number of faulty products in the sample, α = user defined constraint limit.

If the Filtration Index for a variable is close to zero, it can be considered that the variable is equally distributed among good and faulty products and hence, should not be considered for analysis. For example, if a dataset has 50 good products and 25 faulty products and among them 10 good products and 5 faulty products have the same variable, then FI will be [(10/50)-(5/25)] = 0. As the variable is similarly distributed among the good and faulty products, it will have no effect on clustering. However, it is not expected that a variable would have FI value exactly zero. In practice, the user has to define the value (or range of values) for α depending on the characteristics of the dataset. Although the FI has been proposed for categorical variables, it can similarly be used for numerical variables.

An improved GSOM process taking clustering quality and FI into consideration is described below:

1. Initialize the weight vectors of the starting nodes (usually four) with random numbers between 0 and 1.
2. Calculate the growth threshold (GT) using equation (3)
3. Determine the winning neuron using equation (1).
4. Update the weight vectors of the neighbourhood using equation (2).
5. Calculate error values and if the error of node i is greater than GT, grow nodes if i is a boundary node. Distribute weights to neighbors if i is a non-boundary node.
6. Initialize the new node weight vectors to match the neighboring node weights.
7. Repeat steps 3 – 6 until all inputs have been presented and node growth is reduced to a minimum level.
8. Reduce learning rate and fix a small starting neighborhood.
9. Find winner and adapt the weights of the winner and neighbors in the same way as in growing phase.

The GSOM process with clustering quality improvement described above can be summarized as follows. Primary or basic data is first pre-processed. Then the dataset is filtered to remove insensitive variables. If there are some categorical data in the dataset, these are converted to numerical values Data are then normalized to feed into the GSOM. The quality of the cluster maps are then evaluated using equation (7). If the clustering quality is not acceptable, the dataset is further refined by removing insensitive data. Finally the cluster maps are analyzed and interpreted with the knowledge of the dataset used.

3. Results and Discussions

The method proposed has been applied to a complex manufacturing datasets. The results demonstrate the effectiveness of the methodology. Details of the simulation results are described in the following sections.

Recorded manufacturing process data can be used to analyze the performance of a process. However, it is difficult to find the causes of any abnormal output or the factors resulting in lower yield rates [11]. To determine whether GSOM could deal with a complex manufacturing dataset, a simulation has been run with a large dataset obtained from Motorola USA. The quality problem of the Motorola wafer fabrication process is described in reference [1].

In the context of analysis of manufacturing quality problems, the focus involves two main aspects - separation of good and faulty products and identifying the reason for yield failure. The present study focuses on the first aspect. The challenge is not solely in clustering, but also to obtain a meaningful and adequate number of clusters. With meaningful clusters, grouped in appropriate numbers, identification of the reasons that contribute significantly to the differentiation of clusters should become a simpler task.

The dataset is the historical wafer data collected for 2500 wafers over a 2-month period. The input database measured 133 parameters by 16,381 entries organized into an Excel file. The data consisted of wafer probe data of 39 wafer probe functional tests, process control data (59 numerical electrical PC measurements probed at 8 sites per wafer) and process step data such as material vendor/lot, wafer boat position, etc. A sample of the dataset is shown in Table 1. The second column in the table is the product ID, the 3rd column is the product reference number, and columns C1 to X133 are measured parameters. In the original dataset, there are 59 'C' columns (C1-C59), 38 'K' columns (K60-K98), and 34 'X' columns (X99-X133). The reference number in the 3rd column identifies which product is good and which product is faulty. Reference numbers above 8750 indicate good products and below 8750 indicate faulty product.

As shown in Table 1, there are many categorical variables especially in the X-columns. This dataset must be converted into a suitable format to use in GSOM. Pre-processing of data includes removal of outliers, transformation of categorical data into numerical values, and normalization. It was found that there are some excessively large values in the dataset and if these are not removed, most of the values will become zero after normalization. After removing the outliers, the categorical data are transformed into binary. Considering the complexity of the dataset, about one quarter of the entries (4000) was considered for simulation purposes.

No.	NAME	REF	C1	C2	•••	K60	K61	•••	X132	X133
1	J546040_12_1	9628	1.00E-09	2.21E-02	•••	33	18	•••	R2793	Dec-18-95
2	J546040_12_2	9628	-1.33E-09	2.39E-02	•••	33	18	•••	R2793	Dec-18-95
3	J546040_12_3	9628	8.33E-10	2.31E-02	•••	33	18	•••	R2793	Dec-18-95
4	J546040_12_4	9628	-4.77E-10	2.36E-02	•••	33	18	•••	R2793	Dec-18-95
•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•
16381	F606617_21_5	9584	-2.69E-08	2.23E-02	•••	19	18	•••	R285	Feb-26-96

Table 1. Motorola Wafer Manufacturing Data showing dimensionality and layout)

SF(IF≤ 0)	**0.1**	0.5	0.7	0.8	0.9	**0.95**
CQ	**0.25**	0.40	0.45	0.49	0.51	**0.52**
IF (SF=0.95)	**≤ 0.05**	≤ 0.10	≤ 0.12	≤ 0.13	≤ 0.14	**≤ 0.15**
CQ	**0.71**	0.76	0.78	0.79	0.80	**0.81**

Table 2. Improvement of Clustering Quality Changing SF and IF

Simulation was run at SF 0.1 first but a very poor (0.25) CQ was obtained. Clustering quality was improved with higher SF but the maximum CQ was only 0.52, obtained at an SF of 0.95. It is thought that the large number of attributes imposed unnecessary noise in the dataset. It may be necessary and of great insight to study the effect of removing several attributes from the input dataset and to check the impact on the generated maps.

The categorical data from the dataset were reduced using equation (8). First using the constraint IF ≤ 0.05, 163 categorical variables were deleted and the simulation was rerun. A significantly higher CQ of 0.71 was achieved. To test further, more categorical variables were deleted. Using IF ≤ 0.15 a CQ of 0.81 was achieved. As further deletion of variables did not produce any more significant improvement, no further deletion was done. Moreover, if many variables are deleted, some important features may be lost. A summary of the results is presented in

Table 2 and a cluster map of SF=0.95 and IF ≤ 0.15 is presented in Figure 3.

Although higher clustering quality is desired, it may not be possible to obtain high CQ with complex manufacturing data as 100% separation of all good and faulty products is not possible. For example, consider that the desired dimension of a product is 10mm. If the dimension of the finished product is 11mm it will be considered a failure and if the dimension of final product is 15mm it is also a failure but certainly different to the previous failure. It cannot be expected that all products with a dimension 10mm (plus tolerance) be clustered in one group and the rest will be grouped in another cluster. There will certainly be some mixing. In practice, some clusters with 100% good and faulty products and the rest of the clusters with majority either good or faulty products are expected. In this study some clusters with 100% good and faulty products were obtained and other clusters either contained majority of good products or faulty products.

4. Conclusions

The proposed methodology is a significant contribution
to the data mining techniques available, especially in dealing with complex manufacturing datasets. The GSOM algorithm has been proven to be an efficient algorithm to analyze unsupervised DNA data. However, it produced unsatisfactory clustering when used on manufacturing data. Moreover, there was no benchmark to monitor improvement in clustering. The present study has proposed methods to evaluate quality of the clusters produced by GSOM and to remove insignificant variables from the dataset. With the proposed modifications, significant improvement in unsupervised clustering was achieved with simple as well as complex manufacturing data. Possible further improvement in cluster analysis may be achieved by applying evolutionary optimization algorithms.

For manufacturing data, the objective is not limited to finding the separation of good and faulty

products. The main objective is to find the underlying reason for poor yield. To discover this, it is necessary

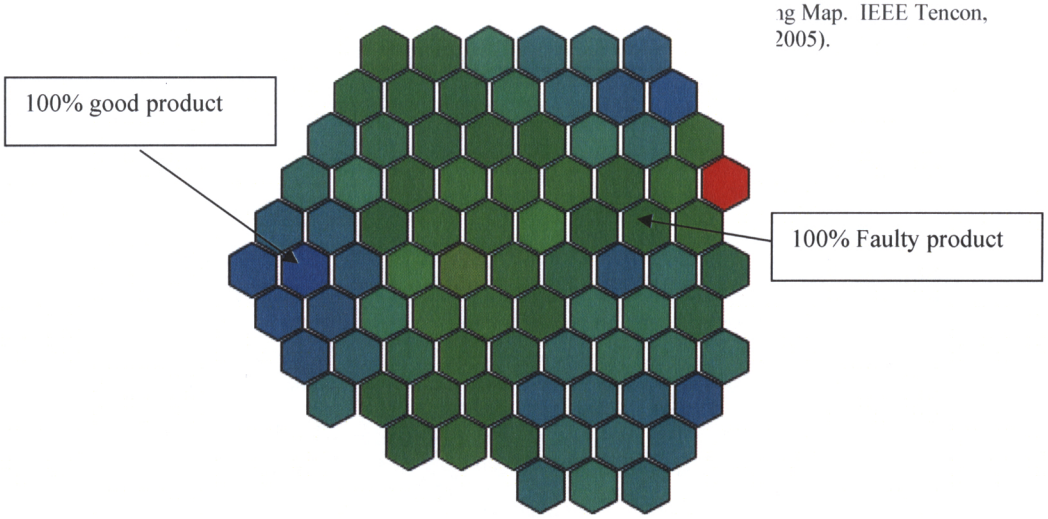

Figure 3. Cluster Map of Motorola Wafer Data (SF: 0.95 and IF=≤ 0.15)

to create clusters of good and faulty products, as has been done in this study. The technique is being extended to model the failure causes of the lower yielding products.

Acknowledgements: The authors acknowledge the significant contribution of Professor Saman Hagamuge and Dr Alan Smith of University of Melbourne to this research work. The authors would also like to thank Robert M. Gardner, Director Intelligent Systems Research Lab, Motorola USA, for kindly providing the valuable semiconductor wafer data.

References

1. Gardner, M., Bieker, J., Elwell, S., Thalman, R., Rivera, E.: Solving Tough Semiconductor Manufacturing Problems using Data Mining. IEEE/SEMI Advanced Semiconductor Manufacturing Conference, Boston, MA (2000) 46-55.
2. M. A. Karim, A. J. Smith, M. Islam and S. Halgamuge, A comparative study of manufacturing practices and performance variables, *International Journal of Production Economics (ISI impact factor-1.183),* 112 (2008), 841–859
3. M. A. Karim, A. J. Smith and S. Halgamuge, Empirical relationships between some manufacturing practices and performance, *International Journal of Production Research,* Vol. 46, No. 13, 1 July 2008, 3583–3613.
4. Russ, G., Karim, M. A., Islam, A., Hsu, A. L., Halgamuge, S. K., Smith, A. J. R, Kruse, R.: Detection of Faulty Semiconductor Wafers using Dynamic ıg Map. IEEE Tencon, 2005).
5. Hsu, A. L. (2005). Interactive data mining with dynamic self-organising maps, Ph.D. Thesis, The University of Melbourne, Melbourne.
6. Kohonen, T.: Self-Organising Maps. Springer , Berlin (1997).
7. Alahakoon, D., Halgamuge, S. K., Srinivasan, B.: Dynamic Self-Organising Maps with Controlled Growth for Knowledge Discovery. IEEE Transactions on Neural Networks. 11 (2000) 601-614.
8. Alahakoon, D.: Controlling the Spread of Dynamic Self Organising Maps. Journal Neural Computing and Applications, 13(2), (2004) 168-174.
9. Hsu, A. L., Tang, S.-L., Halgamuge, S. K.: An Unsupervised Hierarchical Dynamic Self-Organizing Approach to Cancer Class Discovery and Marker Gene Identification in Microarray Data. Bioinformatics 19 (2003) 2131–2140.
10. Vesanto, J. (2000). Using SOM in Data Mining. *Masters thesis*, Helsinki University of Technology.
11. Hou, T.-H., Liu, W.-L., and Lin, L.: Intelligent Remote Monitoring and Diagnosis of Manufacturing Processes Using an Integrated Approach of Neural Networks and Rough Sets. Journal of Intelligent Manufacturing 14 (2003) 239-253.