Expert Systems with Applications 36 (2009) 10158–10167

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Wafer defect pattern recognition by multi-class support vector machines by using a novel defect cluster index

Li-Chang Chao [a,b,*], Lee-Ing Tong [a]

[a] *Department of Industrial Engineering and Management, National Chiao Tung University, 1001 Dah-Hsei Road, Hsin-Chu 300, Taiwan, ROC*

[b] *Department of Industrial Engineering and Management, Diwan University, No. 87-1, Nanshi Li, Madou Town, Tainan County 72153, Taiwan, ROC*

ARTICLE INFO

Keywords:
IC
Cluster index
Defect pattern
Support vector machines

ABSTRACT

Wafer yield is an important index of efficiency in integrated circuit (IC) production. The number and cluster intensity of wafer defects are two key determinants of wafer yield. As wafer sizes increase, the defect cluster phenomenon becomes more apparent. Cluster indices currently used to describe this phenomenon have major limitations. Causes of process variation can sometimes be identified by analyzing wafer defect patterns. However, human recognition of defect patterns can be time-consuming and inaccurate. This study presents a novel recognition system using multi-class support vector machines with a new defect cluster index to efficiently and accurately recognize wafer defect patterns. A simulated case demonstrates the effectiveness of the proposed model.

1. Introduction

As an important index for evaluating the production efficiency of integrated circuits (IC) manufacturers, wafer yield refers to the probability that a chip on a wafer is defect-free. Defects are physical anomalies which result in circuit faults. A common method of monitoring wafer yield is on-line defect detecting in IC manufacturing. Wafers are inspected during manufacturing by retrieving information about defect patterns by manually inspecting or automatically classifying defects (Merino, Cruceta, Garcia, & Recio, 2000). Each of these patterns is associated with a well known manufacturing problem and can provide process and product engineers with valuable clues for identifying the underlying cause and thereby improve yield (Friedman, Hansen, Nair, & James, 1997).

Stapper (1985) indicated that defects are typically clustered rather than dispersed randomly over a wafer, and these clusters become more evident as wafer size increases. Many cluster indices have been developed to depict the intensity of defects scattered on a wafer (Stapper, 1973; Tyagi & Bayoumi, 1992, 1994). The negative binomial yield model (Stapper, 1973) utilizes a cluster parameter α to evaluate the intensity of defects clustered. Cluster parameter α can be quite scattered and sometimes negative (Cunningham, 1990). Tyagi and Bayoumi (1992, 1994) proposed a variance/mean ratio V/M to measure the intensity of defects clustered. The values of V/M depend on how the grids are selected and cannot indicate the gradualness of cross-wafer defect density variations (Tyagi & Bayoumi, 1992, 1994). Jun, Hong, Kim, Park, and Park (1999) proposed a cluster index *CI* to evaluate the intensity of defects clustered on a wafer. In some cases, *CI* values calculated from different defect patterns may be similar.

Constructing a wafer defect recognition system is a very important issue in IC manufacturing. Three fundamental approaches to solving pattern recognition problems are statistical approach, heuristic approach and simulation approach (Nieddu & Patrizi, 2000). The statistical approach classifies patterns based on an extracted features set and an underlying statistical model for generating these patterns. The heuristic approach utilizes soft computing schemes such as genetic algorithms and fuzzy sets to perform pattern recognition. However, genetic systems typically require expensive evaluation processes to achieve optimal solutions (Bhanu, Lee, & Ming, 1995). Further, a major limitation of the fuzzy logic controller is that the linguistic control rules are hard to generate. The fuzzy logic controller also requires knowledge and experience of human experts (Chen & Chang, 1998). The simulation approach emulates the computational paradigm of a biological system, subsequently leading to a class of artificial neural systems termed neural networks (Jain, Duin, & Mao, 2000; Nieddu & Patrizi, 2000). However, a major limitation of neural networks is their inability to determine the number of layers and number of neurons per layer (Fiesler, 1994).

Support vector machines (SVMs) have been widely used for pattern recognition in recent years. Several studies report that the SVM classification is more accurate than existing classification algorithms (Hsu & Lin, 2002; Joachims, 1998). The SVM has proven

* Corresponding author. Address: Department of Industrial Engineering and Management, National Chiao Tung University, 1001 Dah-Hsei Road, Hsin-Chu 300, Taiwan, ROC. Tel.: +886 35 731896; fax: +886 35 733873.

E-mail addresses: lichang.iem91g@nctu.edu.tw, fredchao@dwu.edu.tw (L.-C. Chao).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.01.003

very effective for pattern recognition (Burges, 1998). Zhou, Su, Jiang, Deng, and Li (2007) presented a novel face and fingerprint authentication system based on multi-route detection. Fusion of face and fingerprint recognition by SVM improved authentication accuracy. Nemmour and Chibani (2006) applied SVM for detecting land cover changes. An SVM-based change detection approach has also been used for mapping urban growth. A combination framework was then used to improve change detection accuracy. Jiang, Huang, Ye, and Gao (2006) proposed an SVM scheme for selecting traditional Chinese paintings from general images and categorizing them as Gongbi (traditional Chinese realistic painting) or Xieyi (freehand style). David and Lerner (2005) presented an SVM-based image classification system for genetic syndrome diagnosis. In the David system, a percentage of the miss-classified patterns are rejected to reduce the expected risk by thresholding the distance of patterns from the hyperplane separating the classes.

Given the limitations of previous cluster indices for wafer defects and the importance of an accurate wafer defect recognition system to the IC industry, this study presents a novel recognition system using multi-class support vector machines with a new defect cluster index. A simulation demonstrates the effectiveness of the proposed model. The new defect cluster index is compared with three existing cluster indices, and the recognition system is compared with existing neural network-based recognition systems.

2. Related literature

This section surveys pertinent literature. The defect cluster indices which are utilized to depict the intensity of defects clustered on a wafer are introduced. Approaches to solving pattern recognition problems are then surveyed.

2.1. Defect cluster index

The intensity of defects clustered on a wafer can be depicted by a defect cluster index. The cluster parameter α of the negative binomial model, the variance/mean ratio V/M and the non-parameters assumption cluster index CI are commonly used. The negative binomial yield model is as follows:

$$Y = \frac{1}{(1 + \bar{\lambda}/\alpha)^{\alpha}} \tag{1}$$

where parameter α, the cluster parameter, is calculated as

$$\alpha = \frac{\bar{\lambda}^2}{(\sigma^2 - \bar{\lambda})}, \tag{2}$$

$\bar{\lambda}$ is the mean number of defects per chip, and σ^2 is the variance in defects per chip. Earlier reports show that cluster parameter α in the negative binomial model may be quite scattered and may even have a negative value when the model is used to predict yield (Cunningham, 1990). Tyagi and Bayoumi (1992, 1994) utilized various grid sizes superimposed on a wafer map to measure the intensity of defects distributed on a wafer. The defects contained within each grid can be used to judge the spatial distribution of defects. The distribution of defects follows a Poisson distribution if the defects are randomly distributed. Because both variance (V) and mean (M) are equal in the Poisson distribution, the value of V/M equals 1 if the wafer defects are randomly scattered. The value of V/M exceeds 1 if the defects distributed on a wafer are clustered. The values of V/M depend on how the grids are selected and cannot indicate the gradualness of cross-wafer defect density variations. Jun et al. (1999) proposed a cluster index based on the projected x and y coordinates of defect locations on a wafer. Defect clustering tends to show clumps in the x and the y coordinates, which result in a large variance in defect intervals. However, showing clumps either on the *x-axis* or on the *y-axis* does not necessarily represent the clustered defects. The clustering index CI can be calculated as

$$CI = \min\left\{\frac{s_v^2}{\bar{v}^2}, \frac{s_w^2}{\bar{w}^2}\right\} \tag{3}$$

where v_i and w_i are a sequence of defect intervals on the *x-axis* and *y-axis* defined as

$$v_i = x_{(i)} - x_{(i-1)}, \quad i = 1, 2, \ldots, n \tag{4}$$

$$w_i = y_{(i)} - y_{(i-1)}, \quad i = 1, 2, \ldots, n \tag{5}$$

where $x_{(i)}$ and $y_{(i)}$ denote the ith smallest defect coordinates on the x *-axis* and *y-axis,* respectively; $\bar{v}$ and s_v^2 represent the sample mean and the sample variance of v_i, respectively; $\bar{w}$ and s_w^2 denote the sample mean and the sample variance of w_i, respectively. The value of CI is close to 1 if the defects are randomly scattered, and the value of CI exceeds 1 if defects are clustered. However, in some cases, the CI of different defect patterns may have similar values. Thus, the intensity of defects clustered on a wafer may be erroneously recognized when using cluster index CI.

In summary, existing wafer cluster indices have the following drawbacks: cluster parameter α of the negative binomial model may be substantially scattered and sometimes negative; the same defect pattern may have different V/M values when the selected grids differ; further, the CI values for different defect patterns may also be similar. Such drawbacks affect performance when these cluster indices are employed to depict defect cluster intensity.

2.2. Recognizing defect patterns

Wafer defect patterns may be random or systematic. Ideally, manufacturing defects should be randomly distributed, and systematic pattern defects should be minimal (Kaempf, 1995). Consequently, the yield loss caused by random defects remains constant (Friedman et al., 1997). Some of the many techniques used for wafer defect pattern recognition are statistical approach, heuristic approach and simulation approach (Nieddu & Patrizi, 2000). The statistical approach classifies patterns based on an extracted feature set and an underlying statistical model for generating these patterns. These features are a set of characteristic measurements extracted from the input data and are used to assign each feature vector to one of c classes. The statistical approach can be viewed as determining a strategy for classifying samples based on the measurement of feature vector, such that classification error is minimized. The heuristic approach attempts to clarify the essential problem and use available personal knowledge to solve it with the assistance of soft computing schemes such as genetic algorithms and fuzzy sets. However, genetic systems usually require the evaluation of numerous candidate solutions. In application domains in which the evaluation process is expensive, the computational effort required to perform numerous evaluations may be prohibitive (Bhanu et al., 1995). Further, a major limitation of the fuzzy logic controller is that the linguistic control rules are hard to generate. The fuzzy logic controller also requires knowledge and experience of human experts (Chen & Chang, 1998). The simulation approach emulates the computational paradigm of a biological system. Current knowledge of cerebral processes is transferred from a neurophysiological medium to an electronic one. This leads to a class of artificial neural systems termed neural networks (Jain et al., 2000; Nieddu & Patrizi, 2000). The networks most frequently utilized to perform the pattern recognition are the feed-forward networks (Jain, Mao, & Mohiuddin, 1996). However, a major drawback of neural networks is the lack of knowledge for determining network topology (number of layers and number of neurons per layer)

(Fiesler, 1994). The radial basis function (RBF) neural network (Duda & Hart, 1973) is a new technique for pattern recognition and has been successful applied to many fields. The RBF neural network can reduce training time and enhance network learning efficiency. Therefore, the RBF network is utilized for comparing with the proposed method in this study.

2.2.1. Radial basis function

The RBF neural network (Duda & Hart, 1973) is a supervised learning network. As Fig. 1 shows, the network architecture consists of three layers: input layer, hidden layer and output layer. The transformation from the input layer to the hidden layer is nonlinear whereas the transformation from the hidden layer to the output layer is linear. The hidden layer is typically a single layer of processing elements, and the number of hidden nodes in the RBF networks is usually determined experimentally. The hidden layer units that use Gaussian transfer functions are called the kernel. The Gaussian transfer function can be described in the following form:

$$h_j = \exp\left[-\frac{(\mathbf{x}_p - \mathbf{c}_j)^T(\mathbf{x}_p - \mathbf{c}_j)}{2\sigma_j}\right] \quad j = 1, 2, \ldots N \tag{6}$$

where h_j is the output of the jth node in the hidden layer, $\mathbf{x}_p$ is the input pattern, $\mathbf{c}_j$ is the center of the Gaussian function for node j, σ_j is the width associated with the Gaussian function of node j, and N is the number of hidden layer nodes.

Training in an RBF network involves finding the centers and widths associated with each kernel function and weights connecting the hidden neurons to the output neurons. The k-means clustering algorithm (Pandys & Macy, 1996) is often used to search for the appropriate center of the kernel function. A node in the hidden layer will produce a great output when the presented input pattern is closed to its center. After the cluster centers of the kernel function are found, appropriate width parameters are selected. These parameters control the extent of overlap in kernel functions. For Gaussian kernel functions, the width parameter represents the standard deviation of the function. The width parameter is most commonly made equal to the average distance between the cluster centers and the training patterns. The learning in the output layer is conducted after the hidden layer learning is complete, and the Least Mean Squares (LMS) algorithm is used to train the RBF network. The LMS algorithm uses the delta rule to adjust connection strengths. Output layer node values are calculated as follows:

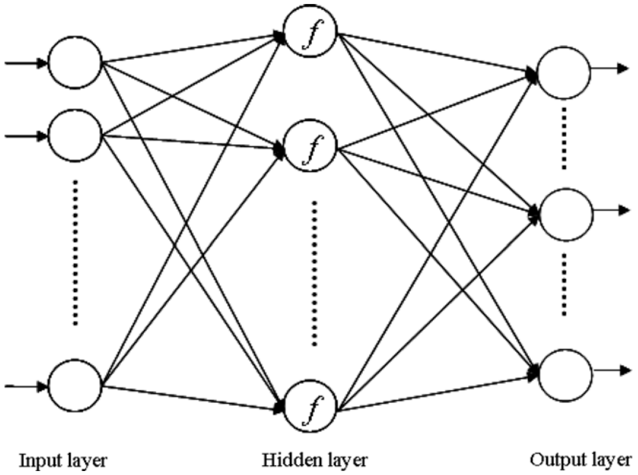


Fig. 1. RBF neural network.

$$o_k = \sum_{j=1}^{N}(w_{jk} \times h_j) \tag{7}$$

where o_k is the output of the kth node in the output layer, w_{jk} is the weight of the jth hidden layer neuron to the kth output layer neuron, h_j is the output of the jth node in the hidden layer, and N is the number of nodes in the hidden layer. The output, o_k, is formed by a weighted linear combination of the output from the hidden layer without nonlinear transformation.

The RBF network is applied to many areas. Su, Yang, and Ke (2002) utilized the RBF network for semiconductor wafer post-sawing inspection. The pros and cons of their technique in comparison with two other inspection methods, visual inspection and feature extraction inspection, were discussed. Doganis, Alexandridis, Patrinos, and Sarimveis (2006) proposed a time series sales forecasting method for short shelf-life food products by combining the RBF network and a specially designed genetic algorithm. The technique was applied successfully to sales data for fresh milk. Sarimveis, Doganis, and Alexandridis (2006) combined the RBF network with fuzzy means algorithm to propose a new classification method. The method is particularly useful for manufacturing processes, particularly in cases where on-line sensors for classifying product quality are unavailable. Yu, Wang, and Lai (2008) utilized the RBF network and the Lagrange multiplier theory to develop a model for solving mean–variance–skewness tradeoffs to optimize portfolio selection.

2.2.2. Support vector machines

The support vector machines (SVM) technique was introduced by Cortes and Vapnik (1995). The original intent of the SVM algorithm was to use a linear separating hyperplane to build a classifier. As Fig. 2 (Schölkopf & Smola, 2002) shows, for all hyperplanes separating data, there exists a unique optimal hyperplane distinguished by the maximum margin of separation between any training point and the hyperplane. The optimal hyperplane is shown as a solid line in Fig. 2.

A training set of instance-label pairs is given as follows:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m), \quad \mathbf{x}_i \in R^n, \quad y_i \in \{+1, -1\}$$

The training set can be divided into two classes by the hyperplane. The hyperplane can be expressed as

$$\langle \mathrm{w}, \mathrm{x} \rangle + b = 0 \tag{8}$$

where ⟨and⟩ are the operators of dot product, **w** is a weight vector orthogonal to the hyperplane, and b is a threshold. The **w** and b are rescaled such that the points closest to the hyperplane satisfy:

$$\langle \mathbf{w}, \mathbf{x}_1 \rangle + b = +1 \tag{9}$$

$$\langle \mathbf{w}, \mathbf{x}_2 \rangle + b = -1 \tag{10}$$

Combining Eqs. (9) and (10) obtains Eq. (11):

$$\left\langle \frac{\mathbf{w}}{||\mathbf{w}||}, (\mathbf{x}_1 - \mathbf{x}_2) \right\rangle = \frac{2}{||\mathbf{w}||} \tag{11}$$

where $||\mathbf{w}||$ is the length of a vector **w**. The distance from the closest point to the hyperplane, called the margin, equals $\frac{1}{||\mathbf{w}||}$. If the optimal hyperplane exists, Eqs. (9) and (10) imply:

$$y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geqslant 1, \quad i = 1, \ldots, m \tag{12}$$

The optimal hyperplane which generalizes well can thus be constructed by solving the following problem:

$$Min. \quad \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2, \tag{13}$$

$$s.t. \quad y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geqslant 1, \quad i = 1, \ldots, m \tag{14}$$

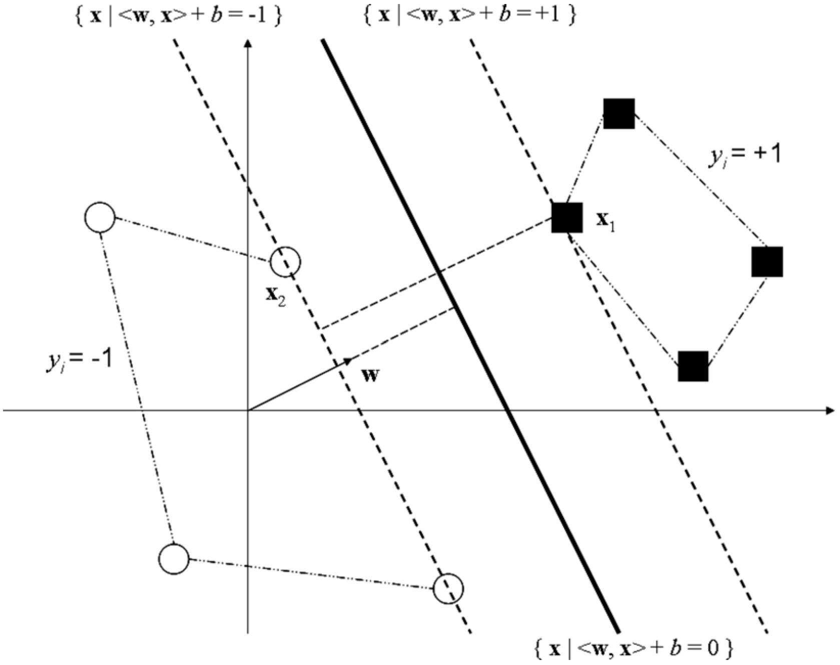


Fig. 2. Optimal hyperplane (Schölkopf & Smola, 2002).

In this case, the dual is more convenient for calculation and is derived by the Lagrangian function as follows:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{1}^{m} \alpha_i (y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1), \tag{15}$$

with Lagrange multipliers $\alpha_i \geqslant 0$. The Lagrange L must be maximized with respect to α_i and minimized with respect to $\mathbf{w}$ and b. Consequently, at this saddle point, the derivatives of L are

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = 0, \tag{16}$$

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \alpha) = 0, \tag{17}$$

Eqs. (16) and (17) get

$$\sum_{1}^{m} \alpha_i y_i = 0, \tag{18}$$

$$\mathbf{w} = \sum_{1}^{m} \alpha_i y_i \mathbf{x}_i. \tag{19}$$

The solution vector thus has an expansion in terms of training examples. The instances $\mathbf{x}_i$, for which $\alpha_i > 0$, are called Support Vectors. Substituting Eqs. (18) and (19) into Eq. (15), the dual form of the optimization problem, gets

$$Max. \quad W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \tag{20}$$

$$s.t. \quad \alpha_i \geqslant 0, \quad i = 1, \ldots, m \tag{21}$$

$$\sum_{1}^{m} \alpha_i y_i = 0, \tag{22}$$

Once α_i is found, the optimal hyperplane can be found. The decision function of the optimal hyperplane can thus be expressed as

$$f(\mathbf{x}) = \left\langle \sum_{1}^{m} \alpha_i y_i \mathbf{x}_i, \mathbf{x} \right\rangle + b \tag{23}$$

To allow the possibility of examples violating Eq. (14), the slack variables $\xi_i \geqslant 0$ are used to relax the separation constraints, and the optimal hyperplane can thus be constructed by solving the following problem:

$$Min. \quad \tau(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i, \tag{24}$$

$$s.t. \quad y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, m \tag{25}$$

$$\xi_i \geqslant 0 \tag{26}$$

where the constant $C > 0$ is the penalty parameter of the error term and determines the tradeoff between margin maximization and training error minimization.

If the training set of instance-label pairs are nonlinearly separable, the linear SVM may not work well again. The nonlinear kernel can then solve the classification problem. The most commonly applied nonlinear kernels are the polynomial kernel, the Gaussian kernel and the sigmoid kernel. The classification problem can obtain reasonable results when the Gaussian kernel is applied to map samples into a higher dimensional space (Keerthi & Lin, 2003). Therefore, the Gaussian kernel selected for sample mapping in this study can be described as

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2), \quad \gamma > 0 \tag{27}$$

where γ is the kernel parameter. The penalty parameter C and the kernel parameter γ are the most critical parameters while the SVM technique is used to classify samples. The grid search in cross-validation can be used to select the best (C, γ) parameter combination. Applying this parameter combination to the training and testing data can enhance generalization accuracy.

2.2.3. Multi-class SVM

The classification problem mentioned above refers to binary classification, in which class labels can only take two values: −1 or +1. Many real-world problems, however, have more than two classes. This study utilizes a multi-class SVM for wafer defect pattern recognition. Methods of multi-class classification are compared.

The one-against-all method (Bottou et al., 1994) constructs m binary SVM classifiers where m is the number of classes. Each classifier is trained to separate one class from the rest, and the ith SVM is trained with all examples in the ith class with +1 labels as well as all other examples with -1 labels. Solving the optimal solution of the one-against-all method can get m decision functions. The unknown $\mathbf{x}$ is in the class with the largest decision function value. The one-against-one method (KreBel, 1999) trains a classifier for each possible pair of classes and constructs $m(m-1)/2$ binary classifiers. The one-against-one method can get $m(m-1)/2$ decision functions when solving the optimal solution. The unknown $\mathbf{x}$ is classified to the class with the highest number of votes. A vote for a given class is defined as the sample frequency attributed by a classifier. The Directed Acyclic Graph solution (Platt, Cristianini, & Shawe-Taylor, 2000) is the same as the one-against-one solution for $m(m-1)/2$ binary SVMs in the training phase. However, in the testing phase, Directed Acyclic Graph method uses a rooted binary directed acyclic graph which includes $m(m-1)/2$ internal nodes and m leaves. Each internal node represents a binary SVM, and each leaf represents a class. The unknown $\mathbf{x}$ starts at the root and moves through a path to either the left node or right node depending on the output value before reaching a leaf node. The unknown $\mathbf{x}$ is then classified to the appropriate class. Vapnik (1998) proposed a method for simultaneously considering all data in multi-class problems by solving a single optimization problem. Solving the optimal solution of the all-data method can get m decision functions. Each decision function can separate one class from the rest. However, the optimal solutions of the m decision functions are not derived from m optimal problems but are obtained by solving one problem. The unknown $\mathbf{x}$ is in the class with the largest decision function value. Because the training time of the one-against-one method is the shortest of these methods (Hsu & Lin, 2002), this method is used for wafer defect pattern recognition in this study.

3. Proposed approach

3.1. A new cluster index

Section 2.1 introduced several defect cluster indices for depicting the intensity of defects clustered on a wafer. This study proposes a new cluster index for depicting the varying intensity of wafer cluster defects. The proposed cluster index CI_E is

$$CI_E(p_1, p_2, \ldots, p_s) = \sum_{i=1}^{s}\left(p_i \times \log_2\left(\frac{1}{p_i}\right)\right) \tag{28}$$

where s represents the number of defect clusters; p_i represents the proportion of defects in the ith cluster to total number of wafer defects. Fig. 3 is the sketch diagram of the proposed cluster index in this study. The more profound the cluster phenomenon, the larger the CI_E, as Fig. 3a1 shows, and vice versa, as Fig. 3b1 and c1 shows. If several wafers have the same total defect number, the more profound the cluster phenomenon, the smaller the CI_E, as Fig. 3a2 shows, and vice versa, as Fig. 3b2 and c2 shows. Clearly, the proposed cluster index CI_E possesses the advantage of accurately detecting the intensity of clustering defects.

3.2. Constructing the defect pattern recognition system

A major cause affecting yield is the degree to which defects are clustered (Friedman et al., 1997; Stapper, Armstrong, & Saji, 1983). In addition to the random pattern in Fig. 4a, common wafer defect clustering patterns include bull's eye pattern, crescent moon pattern, bottom pattern and edge pattern, as Fig. 4b–e shows (Friedman et al., 1997). The specific defect distribution pattern must be determined when wafer yield is medium or low. The manufacturing process can be moderately adjusted by recognizing defect patterns.

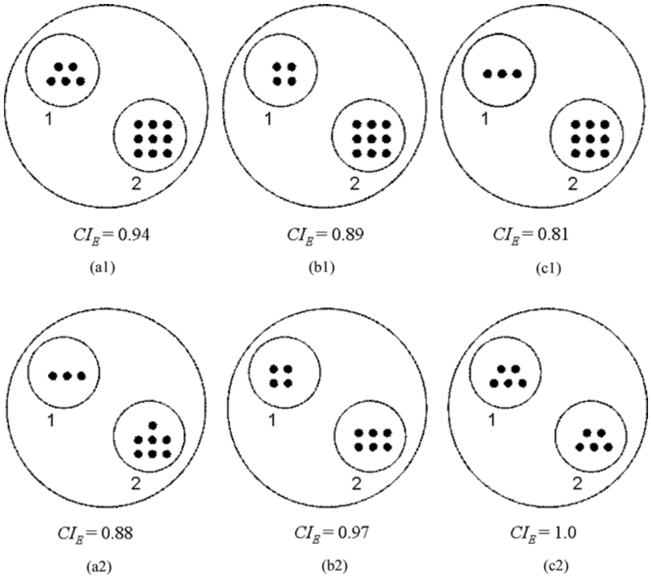


Fig. 3. Sketch diagram of CI_E.

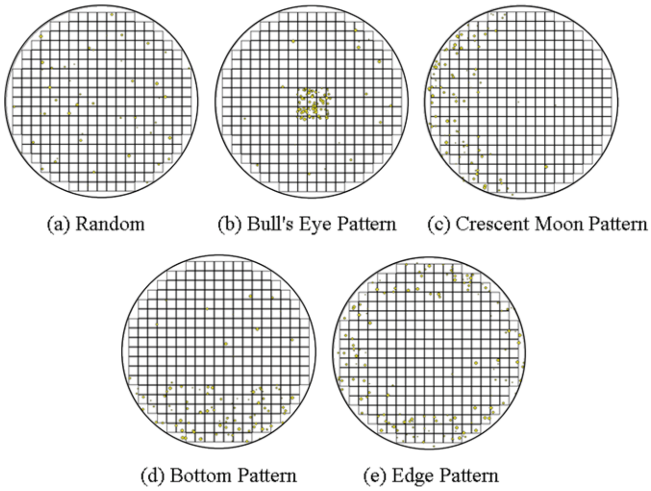


Fig. 4. Defect clustering patterns (Friedman et al., 1997).

Wafers with medium or low yield must be further analyzed to determine whether a specific defect pattern causes the medium or low yield. Therefore, the factors affecting yield are selected as the features for recognizing the five defect patterns.

Yield models are generally a function of D, the average number of defects per unit area, and A, the chip area. Yield models can be described as

$$Y = f(D, A, K) \tag{29}$$

where K represents an empirical correction factor for chip area A (Cunningham, 1990). The average number of defects per unit area D can be used to describe the intensity of the defect-dense areas on a wafer. The average number of defects per unit area D can be used as a feature factor for recognizing defect patterns.

Further, the angle variation CV_A and the distance variation CV_D obtained by measuring the angle variation and the distance variation of the individual defect on a wafer are also utilized as feature factors. Fig. 5 depicts the angle and distance of defects observed in this study. The CV_A and CV_D can be derived as follows:

Step 1: Determine the positive angle θ_i, which is the angle between the coordinates of individual defect and the x-axis. The θ_i can be described as

$$\theta_i = \tan^{-1}\left(\frac{y_i}{x_i}\right), \quad i = 1, 2, \ldots, n \tag{30}$$

where x_i and y_i denote the x and the y coordinates, respectively, of the ith defect in the x–y plant. Sorting θ_i in ascending order obtains $\theta_{(i)}$. A sequence of angle differences is defined as

$$A_i = \theta_{(i)} - \theta_{(i-1)}, \quad i = 1, 2, \ldots, n \tag{31}$$

where $\theta_{(0)} = 0$.

Step 2: Determine L_i as the distance between the individual defect and the origin in the coordinate axes. The L_i can be described as

$$L_i = \sqrt{x_i^2 + y_i^2}, \quad i = 1, 2, \ldots, n \tag{32}$$

Sorting L_i in ascending order obtains $L_{(i)}$. The sequence of distance differences is defined by

$$D_i = L_{(i)} - L_{(i-1)}, \quad i = 1, 2, \ldots, n \tag{33}$$

where $L_{(0)} = 0$.

Step 3: The CV_A and CV_D are defined as

$$CV_A = \frac{S_A^2}{\bar{A}^2} \tag{34}$$

$$CV_D = \frac{S_D^2}{\bar{D}^2} \tag{35}$$

where $\bar{A}$ and S_A^2 denote the sample mean and the sample variance of A_i, respectively, and $\bar{D}$ and S_D^2 denote the sample mean and the sample variance of D_i, respectively. The variations of the angle differences and the distance differences are smaller when defects are randomly distributed than when defects are clustered. One of these two variations is increased regardless of the defect pattern. Therefore, the wafer map presents certain patterns of defect clusters as long as one of these differences posses a large variation. Therefore, CV_A and CV_D can provide feature factors for recognizing defect patterns. Finally, given the superior accuracy of the proposed cluster index CI_E in detecting cluster defect intensity, CI_E is employed as the feature factor for recognizing defect patterns.

Four feature factors (D, CV_A, CV_D and CI_E) are suggested for recognizing defect patterns. A multi-class SVM classifies wafer defect patterns by employing these four feature factors as inputs and one of five defect patterns as output. The relationships between these feature factors and defect patterns can be constructed by presenting the adequate training and testing samples in the multi-class SVM. The multi-class SVM can be used to classify wafers with medium or low yield. Fig. 6 shows the framework of the proposed system for recognizing wafer defect patterns.

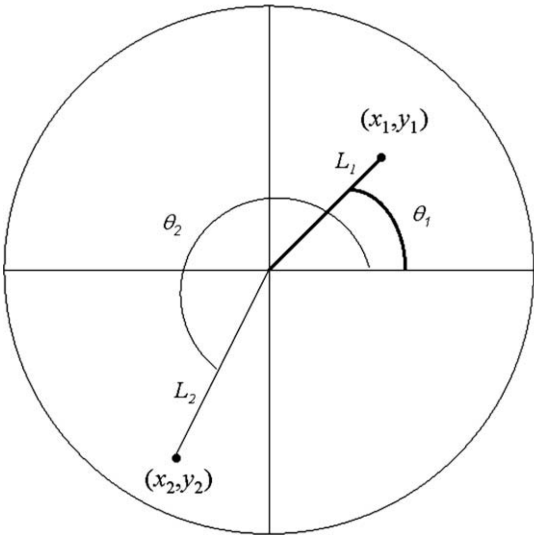


Fig. 5. Angle and distance of wafer defects.

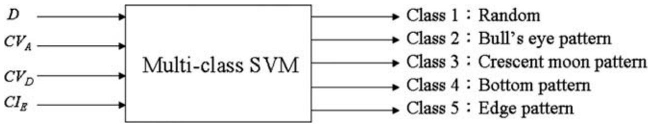


Fig. 6. Recognition system framework.

4. Implementation

4.1. Simulation study

Fig. 4 shows the common wafer defect cluster patterns. In this study, Borland Delphi programming language is employed to simulate various defect cluster patterns in 8-in. wafers. This study employs three design factors to simulate defect cluster patterns: defect number, percentage of defects located in grey regions and size of grey regions. The following briefly describes these three design factors.

(1) *Defect number:* The number of defects distributed over the entire wafer. Five factor levels for 25, 50, 100, 200 and 300 defects are simulated.
(2) *Percentage of defects located in grey region:* The grey region represents the defect-dense areas on a wafer. In the four clustering patterns, four percentages, 80%, 85%, 90% and 95%, of the total number of defects are located in grey regions, and the remaining defects are distributed randomly. Distribution of defects for the four clustering patterns depends on the percentage of defects located in grey regions.
(3) *Size of grey region:* Three sizes of grey regions considered are: 25, 49 and 81 cm^2.

According to the above three design factors, 15 factor-level combinations exhibit random pattern and 60 factor-level combinations exhibit the other four defect patterns. Each trial of factor-level combination is replicated five times, to obtain 1275 simulation trials. Specifically, there are 1275 simulated wafer maps. In order to compare the differences between the proposed cluster index CI_E and the other cluster indices. There are four responses, namely α, V/M, CI and CI_E, are used for each simulation trial. Moreover, to utilize the proposed multi-class SVM for classifying defect patterns and comparing the accuracy of the method with RBF neural network, four feature factors (D, CV_A, CV_D and CI_E) are obtained for each simulation trial by simple calculation.

4.2. Relationship between cluster indices and design factors

The influences of the three designed factors, which are defect number, percentage of defects located in grey region and size of grey region, are analyzed for each defect pattern. The 1275 simulated wafer maps described in Section 4.1 are used to calculate the value of the following four cluster indices: α, V/M, CI and CI_E. The cluster intensity presented on a specific defect pattern is utilized to analyze the effectiveness of those cluster indices.

First, consider five levels for the defect number factor: 25, 50, 100, 200 and 300. The more serious the cluster phenomenon, the larger the proposed CI_E, and vice versa. Fig. 7 depicts the relationship between the cluster indices and defect number for five defect patterns. Fig. 7 shows that cluster parameter α is quite scattered and even negative for some levels, and the other cluster indices can reflect the cluster intensity when the defect number is increased. Further, the cluster index CI cannot clearly reflect the cluster intensity for the crescent moon pattern and the edge pattern. However, the proposed cluster index CI_E can clearly reflect the cluster intensity for all defect patterns.

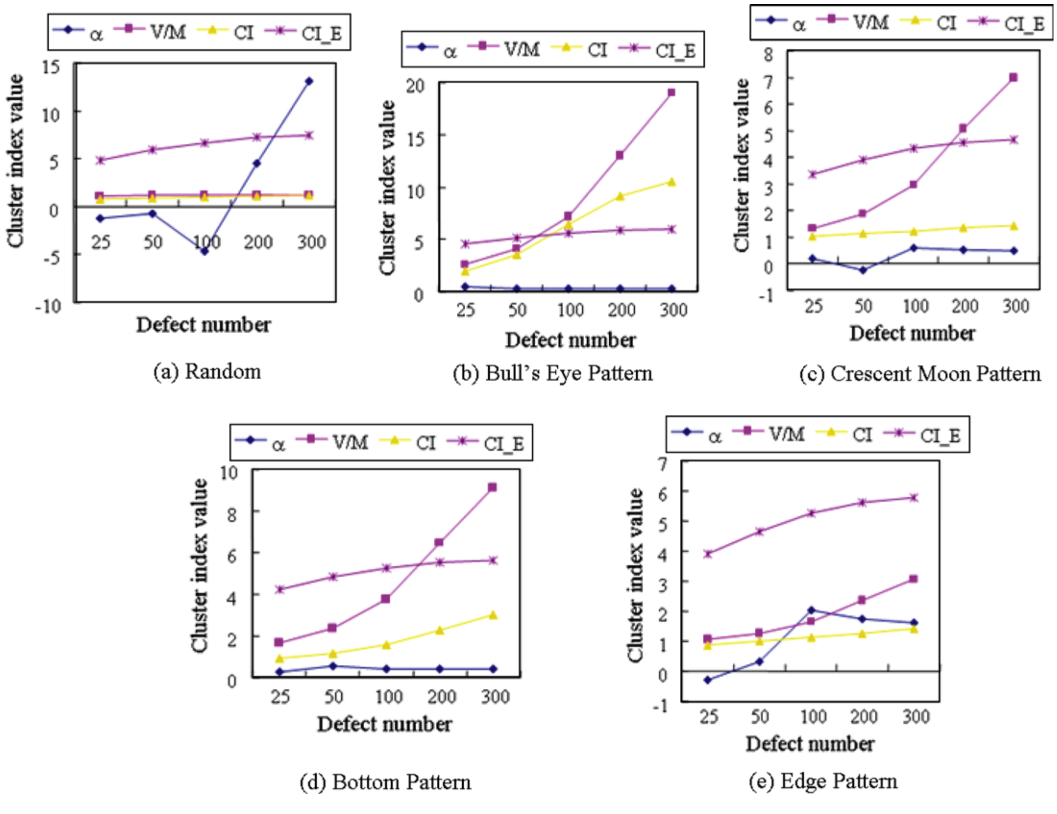
Second, consider four different percentages of defects located in the grey region factor: 80%, 85%, 90% and 95%. The remaining defects are distributed randomly. If several wafers have the same number of total defects, the more serious the cluster phenomenon, the smaller the proposed CI_E, and vice versa. Fig. 8 depicts the relationship between the cluster indices and the four different defect percentages for five defect patterns. Fig. 8 shows that cluster parameter α is quite scattered and even negative for some levels. The cluster index V/M cannot clearly reflect the cluster intensity for the edge pattern. The cluster index CI cannot clearly reflect the cluster intensity for crescent moon pattern, bottom pattern and edge pattern. However, the proposed cluster index CI_E can clearly reflect the cluster intensity for all defect patterns.

Finally, consider three levels for the size of grey region factor: 25, 49 and 81 cm^2. The smaller the size of the grey region, the more serious the cluster phenomenon. If several wafers have the same total number of defects, the more serious the cluster phenomenon, the smaller the proposed CI_E, and vice versa. Fig. 9 depicts the relationship between the cluster indices and the three levels of region size for five defect patterns. Fig. 9 shows that cluster parameter α is quite scattered and even negative. The cluster index V/M cannot clearly reflect the cluster intensity of the edge pattern. The cluster index CI also cannot clearly reflect the cluster intensity of the crescent moon pattern or the edge pattern. However, the proposed cluster index CI_E can clearly reflect the cluster intensity for all defect patterns. The above discussion of cluster indices and these three design factors shows that the proposed cluster index CI_E detects the cluster intensity for the five defect patterns more accurately than the other cluster indices.

4.3. Multi-class SVM wafer pattern recognition

A wafer must be further diagnosed to detect whether there exists a specific clustering pattern of defects when the wafer presents a medium or low yield. To recognize wafer defect patterns, the 1275 simulated wafer maps described in Section 4.1 were utilized as samples for constructing the multi-class SVM. The 1275 wafers were divided into two parts: one part containing 1020 wafers used to train the multi-class SVM; the second part containing 255 wafers used to test the accuracy of the multi-class SVM. Four feature factors, D, CV_A, CV_D and CI_E, are obtained for each simulation wafer by simple calculation. These four feature factors and the respective defect pattern of the 1275 wafer maps are utilized as inputs and outputs for the proposed multi-class SVM. The trained multi-class SVM can then be used to classify wafers presenting a specific defect pattern. Fig. 10 shows some of the defect patterns observed in this study. Cluster intensity is not considered for random pattern. Fig. 10a shows three wafers with 50 defects and three wafers with 100 defects for random pattern. For cluster intensity of 90%, Figs. 10b–10e show three wafers with 50 defects and three wafers with 100 defects for the other defect patterns.

The classification accuracy of the proposed multi-class SVM is compared with that of RBF neural network. The classification accuracy for a specific defect pattern is employed to analyze the performance of these two classification techniques. Software utilized in this study for multi-class SVM and RBF neural network were LIBSVM (Chang & Lin, 2004; Hsu & Lin, 2002) and NeuroSolutions 5.0, respectively. To obtain the generalization results for wafer pattern recognition, fivefold cross-validation was used to determine



(a) Random

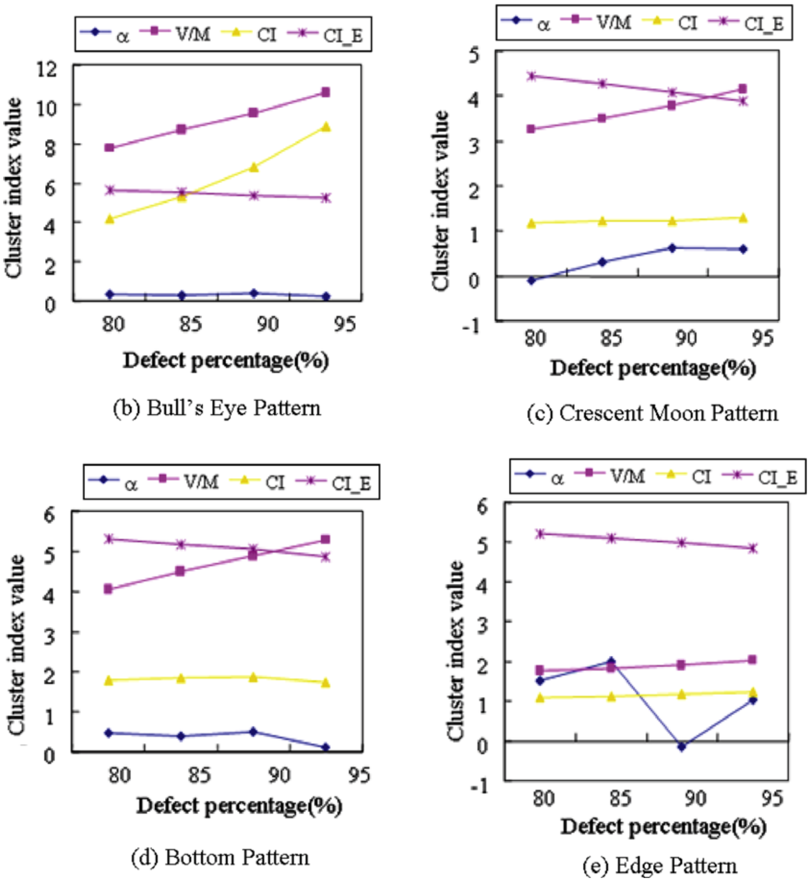
(b) Bull's Eye Pattern

(c) Crescent Moon Pattern

(d) Bottom Pattern

(e) Edge Pattern

Fig. 7. Relationship between cluster indices and defect number factor.



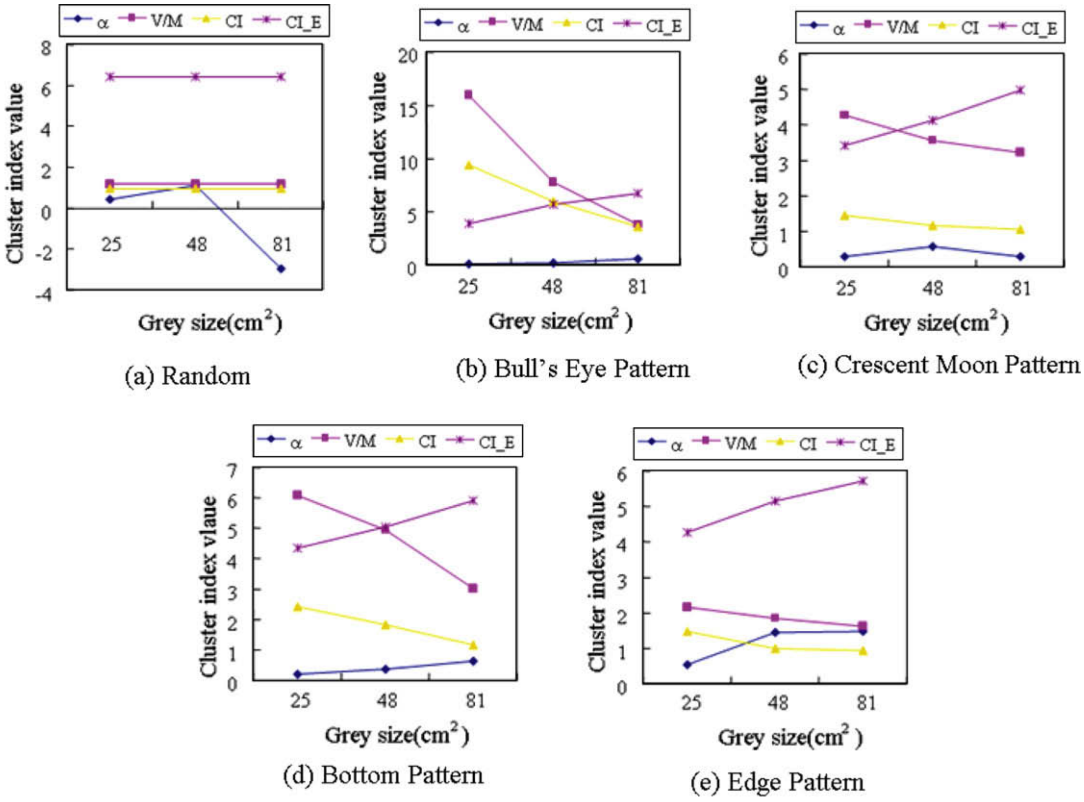
(b) Bull's Eye Pattern

(c) Crescent Moon Pattern

(d) Bottom Pattern

(e) Edge Pattern

Fig. 8. Relationship between cluster indices and defect percentage factor.



(a) Random

(b) Bull's Eye Pattern

(c) Crescent Moon Pattern

(d) Bottom Pattern

(e) Edge Pattern

Fig. 9. Relationship between cluster indices and grey size factor.

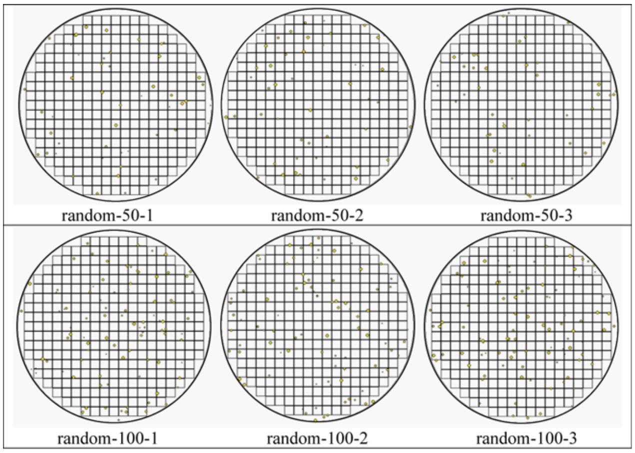


Fig. 10a. Random patterns for defect number 50 and 100.

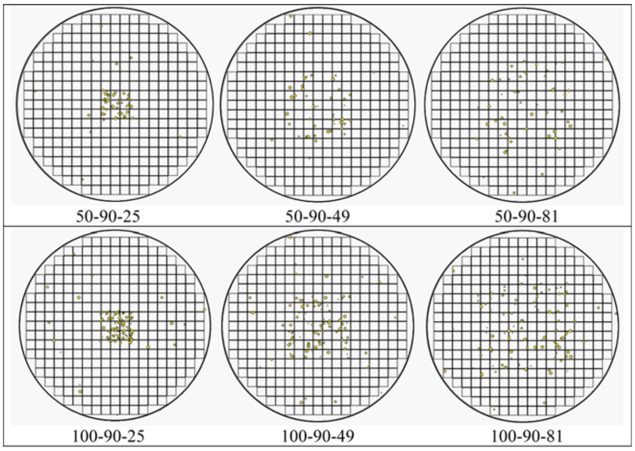


Fig. 10b. Bull's eye patterns with cluster intensity of 90% for defect number 50 and 100.

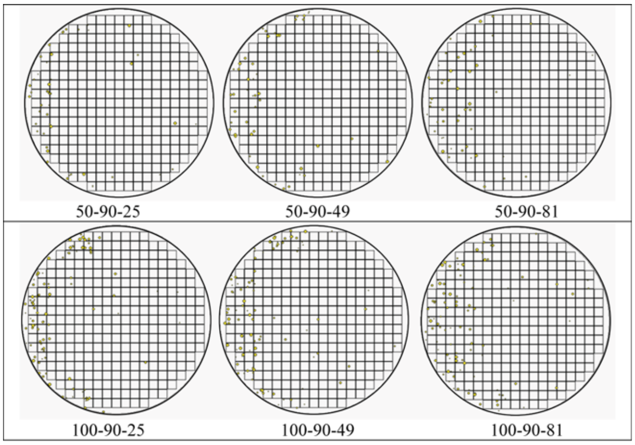


Fig. 10c. Crescent moon patterns with cluster intensity of 90% for defect number 50 and 100.

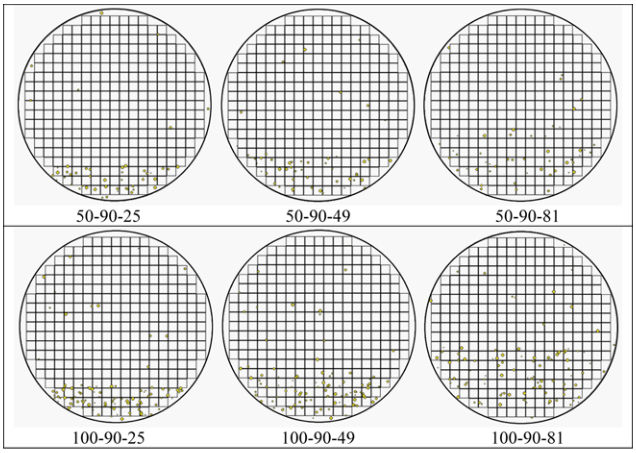


Fig. 10d. Bottom patterns with cluster intensity of 90% for defect number 50 and 100.

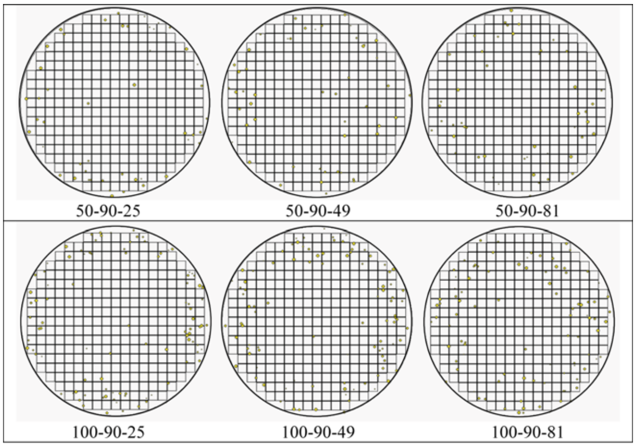


Fig. 10e. Edge patterns with cluster intensity of 90% for defect number 50 and 100.

optimal parameter combinations for these two classification techniques. The penalty parameter *C* and the kernel parameter γ are the most critical parameters for multi-SVM. In this study, the optimal parameter combinations for penalty parameter *C* and kernel parameter γ were 2048 and 0.03125, respectively. Of 255 wafers, 233 could be accurately classified into respective classes. Restated, recognition accuracy when using the proposed multi-class SVM is 91.3725%. Similarly, the same four feature factors and the respective defect pattern of the 1275 wafer were also utilized as the input and output for the RBF neural network. The optimal combination of parameters is obtained by using the following stop criteria: maximum training of 10^5 epochs or mean square error is less than 10^{-6}. The RBF neural network is used to classify wafers which present a specific defect pattern. Of the 255 wafers, 160 can be accurately classified into respective classes. Restated, the recognition accuracy rate achieved by utilizing the RBF neural network is 62.7451%. The classification results for wafer defect pattern recognition shows that the proposed multi-class SVM achieves a more accurate recognition rate than the RBF neural network.

5. Conclusion

As wafer sizes increase, the defect cluster phenomenon increases. Both defect number and defect clustering affect wafer yield. Some proposed defect cluster indices monitor whether a wafer exhibits the cluster phenomenon. The cluster parameter α of the negative binomial model can be quite scattered and sometimes negative. The values of cluster index V/M depend on how the grids are selected. The *CI* values for different defect patterns may also be similar. Among the techniques for recognizing wafer defect patterns, genetic systems usually require expensive evaluation processes to achieve optimized solutions; the linguistic control rules for the fuzzy logic controller are hard to generate; neural networks lack the knowledge for determining the number of layers and number of neurons per layer.

This study presents a novel recognition system that utilizes multi-class support vector machines incorporating a new defect cluster index for recognizing wafer defect patterns. A simulated case is applied to demonstrate the effectiveness of the proposed model by constructing a new defect cluster index and a new recognition system. The new defect cluster index is compared with three existing cluster indices, and the recognition system is compared with one constructed by neural network.

The merits of the proposed approach are summarized as follows:

1. The proposed cluster index CI_E is more accurate than other cluster indices (α, V/M and CI) in terms of detecting the cluster intensity for five various defect patterns.
2. The proposed method utilizes four relevant variables: D, CV_A, CV_D and CI_E as input for constructing the wafer defect recognition system. The classification results for wafer defect patterns show that the proposed multi-class SVM achieves more accurate recognition than RBF neural network.
3. The proposed system can be integrated with KLA inspection machines to recognize wafers presenting medium or low yield to minimize improvement time.

References

Bhanu, B., Lee, S., & Ming, J. (1995). Adaptive image segmentation using a genetic algorithm. *IEEE Transactions on Systems Man Cybernetics, 25*(12), 1543–1567.

Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L., et al. (1994). Comparison of classifier methods: A case study in handwritten digit recognition. In *Proceedings of the international conference on pattern recognition*. Los Alamitos, CA: IEEE Computer Society Press.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*(2), 121–167.

Chang, C. C., & Lin, C. J. (2004). LIBSVM: A library for support vector machines. <http://www.csie.ntu.edu.tw/cjlin/libsvm/>.

Chen, C. L., & Chang, M. H. (1998). Optimal design of fuzzy sliding-mode control: A comparative study. *Fuzzy Sets and Systems, 93*(1), 37–48.

Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning, 20*(3), 273–297.

Cunningham, J. A. (1990). The use and evaluation of yield models in integrated circuit manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 3*(2), 60–71.

David, A., & Lerner, B. (2005). Support vector machine-based image classification for genetic syndrome diagnosis. *Pattern Recognition Letters, 26*(8), 1029–1038.

Doganis, P., Alexandridis, A., Patrinos, P., & Sarimveis, H. (2006). Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering, 75*(2), 196–204.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: John Wiley and Sons.

Fiesler, E. (1994). Comparative bibliography of ontogenic neural networks. In *Proceedings of the international conference on artificial neural networks* (pp. 793–796). Sorrento, Italy.

Friedman, D. J., Hansen, M. H., Nair, V. N., & James, D. A. (1997). Model-free estimation of defect clustering in integrated circuit fabrication. *IEEE Transactions on Semiconductor Manufacturing, 10*(3), 344–359.

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks, 13*(2), 415–425.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1), 4–37.

Jain, A. K., Mao, J., & Mohiuddin, K. m. (1996). Artificial neural networks: A tutorial. *Computer, 29*(3), 31–44.

Jiang, S., Huang, Q., Ye, Q., & Gao, W. (2006). An effective method to detect and categorize digitized traditional Chinese paintings. *Pattern Recognition Letters, 27*(7), 734–746.

Joachims, J. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of ECML-98, 10th European conference on machine learning* (pp. 137–142).

Jun, C. H., Hong, Y., Kim, S. Y., Park, K. S., & Park, H. (1999). A simulation-based semiconductor chip yield model incorporating a new defect cluster index. *Microelectronics Reliability, 39*(4), 451–456.

Kaempf, U. (1995). The binomial test: A simple tool to identify process problems. *IEEE Transactions on Semiconductor Manufacturing, 8*(2), 160–166.

Keerthi, S. S., & Lin, C. J. (2003). Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation, 15*(7), 1667–1689.

KreBel, U. (1999). Pairwise classification and support vector machines. In *Advances in kernel methods: Support vector learning* (pp. 255–268). Cambridge, MA: MIT Press.

Merino, M. A., Cruceta, S., Garcia, A., & Recio, M. (2000). SmartBit™: Bitmap to defect correlation software for yield improvement. In *Advanced semiconductor manufacturing conference and workshop, 2000 IEEE/SEMI* (pp. 194–198). Boston, MA.

Nemmour, H., & Chibani, Y. (2006). Multiple support vector machines for land cover change detection: An application for mapping urban extensions. *ISPRS Journal of Photogrammetry and Remote Sensing, 61*(2), 125–133.

Nieddu, L., & Patrizi, G. (2000). Formal methods in pattern recognition. *European Journal of Operation Research, 120*(3), 459–495.

Pandys, A. S., & Macy, R. B. (1996). *Pattern recognition with neural networks in C++*. CRC Press. pp. 214–230.

Platt, J. C., Cristianini, N., & Shawe-Taylor, J. (2000). Large margin DAGs for multiclass classification. In *Advances in neural information processing systems* (pp. 547–553). Cambridge, MA: MIT Press.

Sarimveis, H., Doganis, P., & Alexandridis, A. (2006). A classification technique based on radial basis function neural networks. *Advances in Engineering Software, 37*(4), 218–221.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT Press.

Stapper, C. H. (1973). Defect density distribution for LSI yield calculations. *IEEE Transaction on Electron Devices (Correspondence), 20*(7), 655–657.

Stapper, C. H. (1985). The effects of wafer to wafer defect density variations on integrated circuit defect and fault distributions. *IBM Journal of Research Development, 29*(1), 87–97.

Stapper, C. H., Armstrong, F. M., & Saji, K. (1983). Integrated circuit yield statistics. *Proceedings of the IEEE, 71*(4), 453–470.

Su, C. T., Yang, T., & Ke, C. M. (2002). A neural-network approach for semiconductor wafer post-sawing inspection. *IEEE Transactions on Semiconductor Manufacturing, 15*(2), 260–266.

Tyagi, A., & Bayoumi, A. M. (1992). Defect clustering viewed through generalized Poisson distribution. *IEEE Transactions on Semiconductor Manufacturing, 5*(3), 196–206.

Tyagi, A., & Bayoumi, M. A. (1994). The nature of defect patterns on integrated-circuit wafer maps. *IEEE Transactions on Reliability, 43*(1), 22–29.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

Yu, L., Wang, S., & Lai, K. K. (2008). Neural network-based mean–variance–skewness model for portfolio selection. *Computers and Operations Research, 35*(1), 34–46.

Zhou, J., Su, G., Jiang, C., Deng, Y., & Li, C. (2007). A face and fingerprint identity authentication system based on multi-route detection. *Neurocomputing, 70*(4–6), 922–931.