Problem Statement-

We are studying wafer dataset which has 6 variables with time varying data. Data is bi-classified as normal and abnormal and thus our main task becomes testing the new time series data as normal or abnormal. We are using MTS approach with 2d-svd algorithm to study the pattern of classification.

**Assumption:**

-Abnormal is codes as 0

-Normal classification is coded as 1

-The data has minimum 104 time frames ( the dataset has been truncated to this value

-The number of training samples is 572

-The number of testing samples is 384

-The data

Data and source-

We have used the wafer dataset :

The wafer database comprises a collection of time-series data sets where each file contains the sequence of measurements recorded by one vacuum-chamber sensor during the etch process applied to one silicon wafer during the manufacture of semiconductor microelectronics. Each wafer has an assigned classification of normal or abnormal. The abnormal wafers are representative of a range of problems commonly encountered during semiconductor manufacturing.

The name of each data file has the following format:

<run>\_<wafer>.<desc>

where

<run> - the run (or batch) number

<wafer> - wafer number within the run

<desc> - file descriptor; set as follows:

6 - data from radio frequency forward power sensor

7 - data from radio frequency reflected power sensor

8 - data from chamber pressure sensor

11 - data from 405 nm emission sensor

12 - data from 520 nm emission sensor

15 - data from direct current bias sensor

ann - descriptive information

Details of pre-processing-

The wafer dataset was divided into two datasets for training and testing purpose. The training dataset has 572 data divided into two folders ( abnormal- 72 , normal – 500 ). The testing dataset has 384 testing samples (Abnormal – 48 Normal -336). None of the parameters of dataset had null values.

Common Terms used in program:

Trainlabel : stores the label of the training dataset

Testlabel : stores the label of the testing dataset

Abnormaltraining : Number of training sets used having classification as Abnormal ( value 0)

Normaltraining : Number of training sets used having classification as normal ( value 1)

Classification algorithm used—

 Wehave used a new approach for MTS classification using two-dimensional singular value decomposition (2dSVD) .  Let **Q** denote the MTS sample, **Q** is stored in an m×n matrix, where m is the number of observations and n   is the number of variables, **u**1 is the first singular vector of **Q**, **u**2 is the second singular vector of **Q**, σ is the vector of the singular values of **Q**T**Q**, and σstd=σ/∣σ∣ is the normalized singular value vector. The first approach (Li’s first) considers //\*\*\*\*\*\*iska dekh lena kaun si wali approach lkhni hai \*\*\*\*\*\*\*????the first singular vector and the normalized singular values, i.e. the first singular vector **u**1 concatenated by the normalized singular value vector σstd. The second approach (Li’s second) takes into account the first two dominating singular vectors weighted by their associated singular values, i.e. the weighted first singular vector w1**u**1 concatenated by the weighted second singular vector w2**u**2, View the MathML source. We refer to the two different feature vector selection approaches as Li’s first and Li’s second approach hereafter.

2dSVD  is an extension of standard SVD. 2dSVD is based on two-dimensional matrices rather than one-dimensional vectors. MTS row–row and column–column covariance matrices are constructed directly using the original MTS samples and their eigenvectors are computed for MTS feature extraction. In low rank approximation, 2dSVD captures explicitly the two-dimensional nature of the two-dimensional objects, such as two-dimensional images, two-dimensional weather maps. We give a brief description of 2dSVD as follows.

Given a MTS dataset View the MathML source, where each MTS sample Ti∈Rm×n. Define the averaged row–row covariance matrix **F** and column–column covariance matrix **G**

equation(1)

View the MathML source

equation(2)

View the MathML source

http://www.sciencedirect.com/sd/blank.gifwhere View the MathML source. Let **U**r contains *r* principal eigenvectors of **F** and **V**s contains *s*principal eigenvectors of **G**, Ur≡(u1,…,ur) and Vs≡(v1,…,vs). The 2dSVD of View the MathML source is View the MathML source, where View the MathML source, Ur∈Rm×r, Vs∈Rn×s, Mi∈Rr×s.**M**i is the feature matrix of MTS sample **T**i. A feature matrix is obtained for each MTS sample by using 2dSVD. Let View the MathML source denote the feature matrix of MTS sample **T**i, i=1,2. The distance between two feature matrices d(**M**1, **M**2) is defined as

equation(3)

View the MathML source

http://www.sciencedirect.com/sd/blank.gif

where ∥•∥ means L2 norm.

Given a MTS sample T∈Rm×n and its reconstructed MTS sample View the MathML source, the squared errors between **T** and View the MathML source is defined as [[2]](http://www.sciencedirect.com/science/article/pii/S0950705108000580#bib2):

equation(4)

View the MathML source

http://www.sciencedirect.com/sd/blank.gif

2D-SVD IN DETAIL-

MTS classification using 2dSVD has three steps. First, we calculate the transformations**U**k and **V**s from the training MTS samples and obtain a feature matrix for each training MTS sample; then the feature matrix of the each testing MTS sample is calculated; finally, the MTS samples in testing dataset can be identified by one-nearest-neighbor classifier with distance [(3)](http://www.sciencedirect.com/science/article/pii/S0950705108000580#fd1). The algorithmic procedure is stated below.

**Algorithm 1.**

Two\_SVD\_1NN(TRAIN, TEST)

*Input:* training dataset, TRAIN, testing dataset, TEST.

*Output:* error rate of classification.

1.

Compute row–row covariance matrix, **F**.

2.

**U**r←r principal eigenvectors of **F**.

3.

Compute column–column covariance matrix, **G**.

4.

**V**s←s principal eigenvectors of **G**.

5.

View the MathML source for View the MathML source.

6.

View the MathML source for ∀TEST(i)∈TRAIN.

7.

Perform one-nearest-neighbor classifier on embedded results {**M**i,**N**i}.

8.

Return the error rate of classification.

Let *M* denote the number of MTS samples in training dataset, *N* denote the number of MTS samples in testing dataset. row–row covariance matrix **F** is m×m matrix, column–column covariance matrix **G** is n×n matrix.

RESULT AND ANALYSIS:

We experimented with different values of r and s . By Hit and Trial Error the maximum accuracy we could achieve was 87.2 % when r =8 and s=3.

We observed that with increase in value of r the error rate decreases and becomes constant (around 0.1) .

