Support Vector Machine and K-Nearest Neighbor Based Microcalcification Classification in a Mammographic CAD System

Sara M. Shawky 1, Nabil A. Lashin 2, Ibrahim El Henawy 3
1,2 Department of Information Technology, Faculty of Computers and Informatics, Zagazig University, Egypt
3 Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Egypt

Abstract--- This paper presents a high accuracy computer-aided system to detect microcalcifications and classify them into benign or malignant. The microcalcifications detection procedure is mainly based on a combination of adaptive histogram equalization, median filtering, morphological operations and thresholding. The contribution of this type of decomposition is denoising and enhancing regions of interests (ROI) containing microcalcifications. Feature extraction is performed on detected microcalcifications and their surrounding tissues using the most well known features in the literature then feature selection is done to reduce these features to the most important features according to the accuracy. SVM and K-NN (conventional, fuzzy and voting) classifiers are used. Our results show that the developed methods are effective for quantifying the classification of benign and malignant microcalcifications with an accuracy of 100%.

Key words: Microcalcifications, K-Nearest Neighbors, Probabilistic Neural Network, Clustering K-means, Support Vector Machine.

1. INTRODUCTION

Early detection and treatment of breast cancer are the most effective methods of reducing mortality. Although independent double reading may improve diagnostic accuracy of radiologists in interpreting microcalcifications (MCs) in mammograms, it is an inefficient and costly approach. Hence, for the last 2 decades there is growing interest in the development of computer-aided detection and diagnosis (CAD) schemes for mammography [1,2,3], which aims to provide radiologists with a valuable “second opinion”. CAD systems use image processing algorithms to detect potential MCs within a mammogram, classifying them as benign or malignant.

1.1 Related Work

Many different techniques were used for detection and classification of microcalcifications. In [4], a Genetic Algorithm (GA) technique is proposed, which is characterized by transforming input images into a feature domain, where each pixel is represented by its mean and standard deviation inside a surrounding window of size g×g pixel. A neural-genetic algorithm is proposed and investigated in [5] for feature selection in conjunction with neural and statistical classifiers to classify microcalcification patterns in digital mammograms. Both the wavelet coefficients and the statistical measures of different wavelet detail levels were used in [6] as features that describe effectively any normal and abnormal region. A set of 88 features to differentiate between malignant and benign MCs using K-NN for classification is presented in [7]. Support Vector Machine (SVM), Kernel Fisher Discriminant (KFD), and Relevance Vector Machine (RVM) were considered in [8] as classifiers. A Discrete Wavelet Transform with biorthogonal spline filters was considered in [9]. Recently, a combination of morphological operations for detection of MCs and discriminant analysis and artificial neural networks for classifying them into benign or malignant has been proposed in [10].

1.2 CAD systems

A typical definition of the Computer Aided Diagnosis (CAD), found in literature [11, 12] (Fig.1), can be: A diagnosis made by a radiologist using the output of a computerized scheme for automated image analysis as a diagnostic aid. CAD systems use image processing algorithms to detect potential microcalcifications within a mammogram, classifying them as benign or malignant.

![Fig. 1: A typical CAD system for microcalcification detection and classification](image-url)
2. METHODOLOGY

2.1 The database

The database used in this work was taken from the Mammographic Image Analysis Society (MIAS). The size of all the images is 1024 x 1024 pixels. The existing data in the collection include the coordinates of the centre (x, y) and the radius of the microcalcifications [13]. This paper has used the 12 digitized films containing malignant MCS and the 12 digitized films containing benign MCs, available in the database. These images include radiologist’s “truth”-markings on the locations of any abnormalities that may be presented.

2.2 ROI Specification

The Region of Interest (ROI) was manually selected to contain the areas of abnormalities marked by the radiologist on the mammograms of the MIAS database.

2.3 Microcalcification Detection

Before carrying out the detection step, a mammogram must undergo pre-processing to remove the artefacts and smooth the image (See Fig.2).

A modification of a similar procedure in [14], which is essentially based on top-hat transformations of the image, is used by using an adaptive histogram equalization ([10] and the references cited therein) to better enhance the positions of MCs and reduce false positives (FPs) in subsequent steps, and applying a median filter of size 5 to the adaptively enhanced ROI. The size of the filter was experimentally found to achieve the best results. Fig. 3 shows the resulted images after each step of the procedure in Fig.2.

Fig. 3 Original and processed images used in MCs extraction (a) original ROI (b) The ROI after applying AHE. (c) The Image after applying initial contrast manipulation. (e) The image after applying Top Hat Transformation. (f) The image after squaring twice. (g) the image after applying normalization. (h) The image after applying opening, final contrast manipulation and thresholding processes to (g).

The segmented MCs (Fig. 4 b) were then excluded from the original ROI (Fig. 4 a), providing the surrounding tissue ROI (ST-ROI) in (Fig. 4 c.)

2.4 Features extraction

The features extraction step is the most critical in the context of MCs. Feature extraction involves absorbing all possible information in the image about MCs and their surrounding tissues. Features are classified according to the image characteristics [15, 16, 17, and 18]:

1- Features extracted directly from segmented MCs, such as Perimeter, area, compactness, elongation, eccentricity, thickness, orientation, direction, line, background, foreground, distance and contrast.
2- Features extracted from spatial Gray Level Cooccurrence Matrices (GLCMs): angular second moment, contrast, correlation, variance, inverse difference moment, sum difference average, sum variance, sun entropy, entropy, difference variance, difference entropy, correlation measure 1, correlation measure 2, maximum, average, and area.
3. Feature extraction from the Run Length Matrices (GLRLMs): short runs emphasis (SRE), long runs emphasis (LRE), grey level non-uniformity (GLNU), run percentage (RP), run length non-uniformity (RLNU), low grey level run emphasis (LGLRE), and high grey level run emphasis (HGLRE).


5. First-Order Statistics (FOS): Mean standard deviation, skewness, kurtosis, uniformity, and smoothness.

6. Features introduced in the past few years as modification of classical features: modified skewness, modified standard deviation, modified entropy, and modified energy.

7. Features based on different types of entropies: new entropy, new sum entropy, new difference entropy, and Havrda entropy.

### 2.5 Feature selection

Feature selection is an important part before any classification scheme. The success of a classification scheme largely depends on the features selected. The objective of performing feature selection is three fold:

(a) Improving the prediction performance of the predictors.
(b) Representing highly correlated features by only a single feature, and
(c) Providing faster and more cost effective predictors.

A new measure was proposed and implemented by the same authors [10], using a score based on the absolute difference between the means of the cases in the two groups (benign, malignant) normalized by the pooled standard deviation of the two groups. For the purpose of completeness, this measure is displayed below. It is denoted by TSQR, for feature j, and is calculated from:

$$TSQR(j) = \left( \frac{\bar{x}_j^{(+)} - \bar{x}_j^{(-)}}{n+1} \right)^2$$

where, $\bar{x}_j^{(+)}$ is the average of the jth feature for the malignant cases ($j=1,...,47$), $\bar{x}_j^{(-)}$ is the average of the jth feature for the benign cases ($j=1,...,47$), $x_{ij}^{(+)}$ is the ith observation of the jth feature (malignant), $x_{ij}^{(-)}$ is the ith observation of the jth feature (benign), $n_+$ is the number of malignant cases, and $n_-$ is the number of benign cases.

To select the best features, the TSQR- score is executed iteratively according to the following procedure:

1. Select a dynamic threshold and apply it over the scores of the features.
2. The most significant features are kept according to the applied threshold value.
3. Present the selected features to the Minitab linear discriminant analysis module and calculate the performance measure (Canonical Correlation or Wilk’s Lambda).

Repeat steps 1 through 3 until the maximum performance is achieved.

Using the above approach, the feature space was reduced from 47 features to the most informative 14 features described in Table 1.

<table>
<thead>
<tr>
<th>Category</th>
<th>No.</th>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segmented MCs</td>
<td>2</td>
<td>Difference Foreground Background</td>
<td>Calculated From the Segmented MCs</td>
</tr>
<tr>
<td>GLCM</td>
<td>16</td>
<td>Correlation</td>
<td>Calculated from the Normalized Co-Occurrence matrix</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>Sum Variance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>Sum Entropy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>Entropy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>Area</td>
<td></td>
</tr>
<tr>
<td>GLRLM</td>
<td>30</td>
<td>Short Run Emphasis</td>
<td>Calculated from the gray level run length matrix</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>Long Run Emphasis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>Run Length Non-Uniformity</td>
<td></td>
</tr>
<tr>
<td>Havrda &amp; Charvat</td>
<td>43</td>
<td>Havrda Entropy</td>
<td>Calculated from the Co-Occurrence Matrix</td>
</tr>
</tbody>
</table>

### 3. CLASSIFICATION

This paper has used the K-Nearest Neighbor (K-NN) (the usual KNN, the voting KNN and the Fuzzy KNN), the Probabilistic Neural Network (PNN), the clustering K-means, the single decision tree, and the Support Vector Machine (SVM) classifiers, to classify the detected MCs into benign and malignant tissues. A brief description of these classifiers is presented below. A detailed exposition of these and other classifiers can be found in [19] and [20].

#### 3.1 K-Nearest Neighbors Algorithm

In this method [21], for each test datum, the Euclidean distances between the test datum and all the training data are calculated, and the test datum is assigned the class label that most of the K closest training data have.

(a) The usual KNN

Let the test datum $x_t$ be represented by the feature vector $[x_1^t, x_2^t, x_3^t, ..., x_n^t]$, where $x_i^t$ denotes the value of the kth attribute of the test datum $x_t$ and $x_i^t$ is the transpose of $x_i$. 
The distance between \( x_i \) and \( x_j \) is defined as
\[
d(x_i, x_j) = \sqrt{\sum_{k=1}^{N} (x_{ik} - x_{jk})^2}
\]

The steps of the algorithm are described by the following pseudo code:

\[
\text{INPUT: (a) Already labelled training data } \{x_i | i=1,2,\ldots,n\}.
\]
\[
\text{(b) The test datum } x.
\]

\[
\text{ALGORITHM: FOR } i=1,2,\ldots \text{ up to } n
\]
\[
\text{Determine the distance between } x \text{ and } x_i.
\]
\[
\text{IF } (i \leq k)
\]
\[
\text{Include } x_i \text{ in the set of } K\text{-nearest neighbors.}
\]
\[
\text{ELSE IF } (x_i \text{ is closer to } x \text{ than any previous nearest neighbor})
\]
\[
\text{Delete the farthest of the } K\text{-nearest neighbors.}
\]
\[
\text{Include } x_i \text{ in the set of } K\text{-nearest neighbors.}
\]
\[
\text{END IF}
\]
\[
\text{END FOR}
\]
\[
\text{FOR } c=1 \text{ to } C
\]
\[
\text{p}_c(x) = \text{(no. of neighbors in class } c)
\]
\[
\text{END FOR}
\]
\[
\text{Crisp class label of } x \text{ is } j
\]
\[
\text{When } p_j = \text{max}\{p_1,p_2,\ldots,p_C\}
\]
\[
\text{OUTPUT: (a) Class label of } x.
\]
\[
\text{(b) Class confidence values } p_c. c.
\]

If \( K=1 \), then the class label of the test datum is equal to the closest training datum. If \( K>1 \), then the class label of the test datum is equal to the class label that most of the neighbors have. If there is a tie, then the tie is resolved arbitrarily.

(b) The Fuzzy KNN

One refinement to the KNN algorithm is to weigh the contribution of each of the K neighbors based on its distance to the test datum. Evidently, the closest neighbor should receive the highest weight. It can be accomplished by the following:
\[
p_c(x) = \sum_{i=1}^{K} \left( \frac{1}{\sum_{j=1}^{K} d(x_{ij})^2} \right) \delta(i, c) \quad \forall c
\]

Here the denominator is used for normalization such that \( \sum_{c=1}^{C} p_c(x) = 1 \) holds. Here \( p_c(x) \) is interpreted as the fuzzy membership function.

(c) The Voting KNN

The Voting k-Nearest Neighbor (k-NN) classifier is nonparametric technique [22], where the inverse distance-weighted voting is used. In this approach, closer neighbors get higher votes. Specifically, the vote of the \( k\) neighbor is defined as:
\[
vote(k) = \frac{1}{d_{k} + 1}
\]

where \( d_{k} \) is the Euclidean distance of the \( k\) neighbor from the test sample. The votes of each class are summed and the test sample is assigned to class with the highest sum of votes. Specifically, the decision function for classification is given by:
\[
\text{Decision} = \sum_{i=0}^{n} \text{vote}(i)_{\text{classA}} - \sum_{j=0}^{m} \text{vote}(j)_{\text{classB}}
\]

where \( n \) and \( m \) are integers ranging from 0 up to \( k \), satisfying the equation \( n + m = k \). In this study, \( k \) ranged from 1 up to 11 neighbors. If \( \text{Decision} \) is greater than zero, the test sample is assigned to class A (malignant); otherwise, the test sample is assigned to class B (benign).

3.2 The Probabilistic Neural Network (PNN)

A probabilistic neural network (PNN) minimizes the expected risk of classifying patterns in the wrong category without any real knowledge of the underlying probability distribution form of the training data. The figure below displays the architecture for a PNN (specht13) that recognizes 2 classes.

![Fig. 5 The architecture of the PNN](image)

The PNN Algorithm
For each feature vectors, we know the class to which it belongs. The following sets up the PNN.

\text{Step 1.} Read in the file of feature vectors and class numbers
\text{Step 2.} Sort these into the K sets where each set contains one class of vectors
\text{Step 3.} For each k
  \begin{enumerate}
  \item define a Gaussian function centered on each vector in set k
  \item define the summed Gaussian output function
  \end{enumerate}
Once the PNN is defined, then we can feed vectors into it and classify them as follows:

Step 1: Read input vector and feed it to each Gaussian function in each class.

Step 2: For each group of hidden nodes, compute all Gaussian functional values at the hidden nodes.

Step 3: For each group of hidden nodes, feed all its Gaussian functional values to the single output node for that group.

Step 4: At each class output node, sum all of the inputs and multiply by constant.

Step 5: Find maximum value of all summed functional values at the output nodes.

3.3 Clustering K-means

A cluster is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters. Clustering means that data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. K-means is an exclusive clustering algorithm [23]. The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.
5. Finally, the algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

\[ J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_{ij} - c_j \|^2 \]

where \( \| x_{ij} - c_j \| \) is a chosen distance measure between a data point \( x_{ij} \) and the cluster centre \( c_j \), is an indicator of the distance of the \( n \) data points from their respective cluster centers.

3.4 Binary Decision Tree

Classification trees are used to predict membership of cases or objects in the classes of a categorical dependent variable [24]. A binary decision tree recursively divides the feature space into two subspaces by selecting a threshold to separate input data into two classes each time. An ordered list of binary threshold operations on the features is organized as a tree. Each node has a threshold associating with one or more features to divide the data into its two descendents. The process stops when it only contains patterns of one class. Comparing with neural networks, the decision tree approach is much simpler and faster. After the mammogram was segmented into regions with different gray levels and features, a binary decision tree is used to classify the MCs into benign and malignant classes.

3.5 The Support Vector Machine Classifier

A simple description of the SVM algorithm is provided as follow [25]: Given a set of \( m \) data points \( \{ x_i, y_i \}, i=1,\ldots, m \) where \( x_i \) is the vector of feature values and \( y_i \) is the target value of the \( i \)th case, in the problem of classifying two groups with \( y_j \in \{-1,1\} \), there exists a hyperplane, which may be expressed as:

\[ y_i ((w\cdot x_i) + b) = 0 \]

Where \( w \) is the weight vector having a unit length laid at right angle with the hyperplane, \( b \) is bias. There are also two boundary hyperplanes (Fig. 5). The vectors (data points) that are closest to the hyperplane are called the support vectors. The other points do not influence the position of the two boundary hyperplanes.

During the training stage, SVMs find the maximum-margin hyperplane between two classes. This is the line (in two dimensions), plane (in three dimensions) or hyperplane (in higher dimensions) that maximizes the distance to the nearest data point, \( \frac{1}{\| w \|} \). To construct the SVM classifier one has to minimize the norm of the weight vector \( w \) under the constraint that the hyperplane divides the data so that all the points with the same label lie on the same side of the hyper plane (Fig. 5). This constraint can be expressed by requiring:

\[ y_i ((w\cdot x_i) + b) \geq 1 \]
Hence, to construct a maximal margin classifier one has to solve the convex quadratic programming problem:

Maximizing separation margin, results in a `Quadratic Programming` optimization problem:

\[
\min_{w,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i \\
\text{s.t. } y_i((w.x_i) + b) \geq 1 - \xi_i, i = 1, \ldots, m
\]

where \( \xi_i \) for \( i = 1,2,\ldots,m \) are non-negative slack variables, and \( C \) is a tuning control parameter. During the experimentation, training (and/or validation) steps, the SVM randomly splits the data into two parts 70% for training and 30% for testing and calculates 10-cross-validation accuracy/mean squared error on them.

4. RESULTS AND DISCUSSION

This section presents the results achieved in this work. We measured, quantitatively, the detection performance of the classifiers by computing the sensitivity and specificity on the data. Sensitivity is the conditional probability of detecting cancer while there is really cancer in the image. Specificity is the conditional probability of detecting benign/malignant MC while the true state of the breast is benign /malignant.

4.1 K-NN classifiers results

Tables 2 and 3 show the results in terms of sensitivity and specificity of the three different K-NN classifiers used in this paper. Comparing the results obtained from the K-NN classifier in this study obtained using the Euclidean distances (ED) with varying values of K between 1- 9, one can conclude the following: All the K-NN classifiers with only the best 14 features (with K= 1 through K=9) gave sensitivity = 100% and specificity = 100%, while the K-NN classifiers using all of the 47 features gave 100% sensitivity for k=1-9. For k=1, 3, and 9 all classifiers gave specificity 100% but only 66.67% specificity for k=5 and 7for the standard K-NN and the voting K-NN.

4.2 The decision tree, clustering means and the PNN classifiers

Table 3: Sensitivity and Specificity of The K-NN Classifiers (The 14 Best Features)

<table>
<thead>
<tr>
<th>k</th>
<th>Standard KNN</th>
<th>Fuzzy KNN</th>
<th>Voting KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sensitivity</td>
<td>Specificity</td>
<td>Sensitivity</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.667</td>
<td>0.667</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.667</td>
<td>0.667</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Table 4: Classification Models (Decision Tree, Clustering Means, and Probabilistic Neural Network (PNN)) Parameters and Output Results

<table>
<thead>
<tr>
<th>Project Parameter</th>
<th>Decision Tree</th>
<th>Clustering Means</th>
<th>PNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of predictor variables</td>
<td>Validation method</td>
<td>Cross Validation LOO</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>LOO</td>
<td>10</td>
</tr>
<tr>
<td>Accuracy</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Specificity</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>A_z</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Validatio n Data Results</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>95.83%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>100.00%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Specificity</td>
<td>91.67%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>A_z</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

A_z: Area under ROC curve

Table 5: Confusion Matrices Resulting from Classifiers

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Training Data</th>
<th>Validation Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Predicted Category</td>
<td>Actual Category</td>
</tr>
<tr>
<td>Decision Tree</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Benign</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>Malignant</td>
<td>0</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clustering Means &amp; PNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Category</td>
</tr>
<tr>
<td>Benign</td>
</tr>
<tr>
<td>Malignant</td>
</tr>
</tbody>
</table>
Conclusion: From Table 5, the decision tree, clustering means and PNN classifiers gave similar results (specificity 100%, sensitivity 100% and accuracy 100%) for the training data. For the validation data, the clustering means and the PNN gave similar results (specificity 100%, sensitivity 100% and accuracy 100%), while the decision tree classifier gave 100% sensitivity, 91.67% specificity, and 95.83% accuracy. The results can also be concluded from the confusion matrices of the three classifiers shown in Table 5. Thus the overall performance of the PNN and the clustering means classifiers are in general better than the decision tree classifier.

4.3 Support vector machine results (using DTREG package version 10.0.0)

The SVM was implemented using STATSOFT version 9. This SVM classifier uses the Sequential Minimal Optimization (SMO) algorithm. Different kernel functions and parameters were experimented with. The kernels included the polynomial, RBF and linear kernels with their different parameters. Initially these kernels and their parameters were compared. However, during the CV process, the best parameters are chosen by nested cross-validation procedures.

To evaluate the performance of SVM, this paper used five objective indexes: accuracy, sensitivity, specificity, positive predictive value, and negative predictive value.

Tables 6, 7, 8, 9 and 10 present the optimal configuration of the SVMs used their input data, their optimal parameter values and their classification output results for both the training and the validation phases.

Table 6: SVM Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target variable</td>
<td>VAR00048</td>
</tr>
<tr>
<td>Number of best predictor variables</td>
<td>14</td>
</tr>
<tr>
<td>Type of model</td>
<td>Support Vector Machine (SVM)</td>
</tr>
<tr>
<td>SVM kernel function</td>
<td>Radial Basis Function (RBF)</td>
</tr>
<tr>
<td>Type of analysis</td>
<td>Classification</td>
</tr>
<tr>
<td>Category weights (priors)</td>
<td>Data file distribution</td>
</tr>
<tr>
<td>Misclassification costs</td>
<td>Equal (unitary)</td>
</tr>
<tr>
<td>Validation method</td>
<td>Cross validation</td>
</tr>
<tr>
<td>Number of cross-validation folds</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 7: Input Data and Optimal Parameters for the SVM Model

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of variables (data columns)</td>
<td>48</td>
</tr>
<tr>
<td>Data subsetting</td>
<td>Use all data rows</td>
</tr>
<tr>
<td>Number of data rows</td>
<td>24</td>
</tr>
<tr>
<td>Total weight for all rows</td>
<td>24</td>
</tr>
<tr>
<td>Rows with missing predictor values</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 8: Output (misclassification using the SVM) with training data followed by validation data

<table>
<thead>
<tr>
<th>Category</th>
<th>Actual</th>
<th>Misclassified</th>
<th>Actual</th>
<th>Misclassified</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (Benign)</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>1 (Malignant)</td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>24</td>
<td>0</td>
<td>24</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 9: Confusion Matrices, Training Data followed by Validation Data

<table>
<thead>
<tr>
<th>Predicted Category</th>
<th>Actual Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Benign)</td>
<td>0</td>
</tr>
<tr>
<td>(Malignant)</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 10: Sensitivity, Specificity, and Area under The ROC Curve using the Best 14 Predictors for Training followed by Validation Data

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Validation Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall Accuracy</td>
<td>100.00%</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>100.00%</td>
</tr>
<tr>
<td>Specificity</td>
<td>100.00%</td>
</tr>
<tr>
<td>Precision</td>
<td>100.00%</td>
</tr>
<tr>
<td>Recall</td>
<td>100.00%</td>
</tr>
<tr>
<td>Area under ROC curve</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Conclusion: The accuracy of SVM for classifying malignancies was 100% (24 of 24); the sensitivity, 100%; the specificity, 100%; the positive predictive value, 100%; and the negative predictive value, 100%. A final remark is that the time taken for classification is 1s vs. 189s taken by the other classifiers. Thus the SVM may be recommended especially if the data is of large scale.
REFERENCES


V. BIOGRAPHIES

Sara Mohammed Shawky is a lecturer with the IT Department, Zagazig University. She has obtained her B. Sc. in 2007, in Computer Science with Honors, from the Zagazig University. She is currently writing her M. Sc. in IT at Zagazig University. Her interests include image Processing, Digital Signal Processing and Logic Design. She has published a paper in a reputable International Journal Conference on IT.

Nabil Aly Lashin has obtained his B.Sc. in Communication Engineering (1993) from Zagazig University, M. Sc. in Communication Engineering (1999) from Cairo University, and his Ph. D. in Electronic Engineering and Computer Science (2005) from the University of Berlin, Germany. He is currently an Assistant Professor of IT, with Zagazig University, Egypt. His interests include Digital Signal Processing, Image Processing and Computer Architecture. He has published several articles on IT and Networking.

Ibrahim El Henawy has obtained his B. Sc. in Mathematics in 1973 from Ain Shams University, Egypt; M. Sc. in Numerical Treatments, from Ain Shams University in 1975, his Ph. D. in Mathematics and Computer Science from the State University of New York in 1983. He is the X-dean of the Faculty of Computers and informatics, Zagazig University. He is currently an Emeritus Professor of Computer Science and Acting Head of the department of Computer Science with Zagazig University, Egypt. His interests include Image Processing, Theory of Algorithms, and Computer Graphics. He has published more than 60 papers in several International Journals.