Breast Cancer Classification using Support Vector Machine and Neural Network

Ebrahim Edriss Ebrahim Ali[1], **Wu Zhi Feng**[2]

[1, 2]School of Information Technology and Engineering, Tianjin University of Technology and Education, Dagu Nanlu Road Tianjin, China

Abstract: ***Breast cancer is one of the most leading causes of death among women. The early detection of abnormalities in breast enables the radiologist in diagnosing the breast cancer easily. Efficient tools in diagnosing the cancerous breast will help the medical experts in accurate diagnosis and timely treatment to the patients. In this work, experiments was carried out using Wisconsin Diagnosis Breast Cancer database to classify the breast cancer as either benign or malignant. Supervised learning algorithm -Support Vector Machine (SVM) with kernels like Linear, and Neural Network (NN) are used for comparison to achieve this tasks. The performances of the models are analyzed where Neural Network approach provides more 'accuracy' and 'precision' as compared to Support Vector Machine in the classification of breast cancer, and seems to be fast and efficient method.***

Keywords: Neural Network, Support Vector Machine, Benign, Malignant.

1. Introduction

Cancer refers to the uncontrolled multiplication of a group of cells in particular location of the body. A group of rapidly dividing cells may form a lump, micro calcifications or architectural distortions which are usually referred to as tumors.

Breast cancer is any form of malignant tumor which develops from breast cells. Breast cancer is one of most hazardous types of cancer among women in the world. The world health organization's International Agency for Research on cancer (IARC) estimates that more than 400,000 women expire each year with breast cancer.

Today, there is an urgent need in breast cancer control and it is achieved primarily by knowing different risk factors. Secondly, there is need to detect this disease in early stage by knowing different symptoms of this disease, so it can be cured.

Breast cancer is mainly of two types: Invasive and Non-invasive. Invasive type is the one in which cancerous cells break through normal breast tissue barriers and spread to other parts of the body. While in non-invasive, cancerous cells remain in a particular location of the breast and do not spread to surrounding tissue, ducts or lobules.

Breast analysis techniques have been improved over the last decade. Number of automated classification systems has been developed over last years. Different techniques have varying results. However, there still are issues to be solved: developing new and better techniques. The comparison between different systems helps us to know better system with high performance; this will assist radiologists to take accurate results regarding the disease.

Radiologists still produces some variation in reading images. So, there is a need for automatic interpretation of images or automated classification system, and for this purpose classifier is required. Nowadays many techniques are used for classification but Neural Network (NN) and Support Vector Machine shows better results in many instances. This paper gives comparative analysis of NN and SVM.

2. Support Vector Machine

Support Vector Machine is a new approach to supervised pattern classification which has been successfully applied to a wide range of pattern recognition problems and it is also a training algorithm for learning classification and regression rules from data. SVM is most suitable for working accurately and efficiently with high dimensionality feature spaces in addition to that SVM is based on strong mathematical foundations and results in simple way and very powerful algorithms.

The standard SVM algorithm builds a binary classifier. A simple way to build a binary classifier is to construct a hyper plane separating class members from non-members in the input space. SVM also finds a nonlinear decision function in the input space by mapping the data into a higher dimensional feature space and separating by means of a maximum margin hyper plane. The system automatically identifies a subset of informative points called support vectors and uses them to represent the separating hyper plane which is sparsely a linear combination of these points. Finally SVM solves a simple convex optimization problem.

This machine is presented with a set of training examples, (xi, yi) where the xi are the real world data instances and the yi are the labels indicating which class the instance belongs to. For the two class pattern recognition problem, yi = +1 or yi = -1. A training example (xi, yi) is called positive if yi = +1 and negative otherwise. SVM construct a hyper plane that separates two classes and tries to achieve maximum separation between the classes. Separating the classes with a large margin minimizes a bound on the expected generalization error. The simplest model of SVM called Maximal Margin classifier, constructs a linear separator (an optimal hyper plane) given by (WTXi - ♈) = 0 between two classes of the examples. The free parameters are a vector of weights W. which is orthogonal to the hyper plane and a threshold value ♈. These parameters are obtained by solving the following optimization problem using

Lagrangian duality

Minimize ½||W||^2 ------------- (1)

Subject to Dii (WTXi - ϒ) ≥1,i=1,…….,I ---- (2)

where Dii corresponds to class labels, which assumes value +1 and –1. The instances with non null weights are called support vectors.

In the presence of outliers and wrongly classified training examples it may be useful to allow some training errors in order to avoid over fitting. A vector of slack variables ξi that measure the amount of violation of the constraints is introduced and the optimization problem referred to as soft margin is given below

Minimize C∑i l =1ǫi + ½||W||^2 -------- (3)

w, γ

Subject to Dii (WTXi - ϒ) ≥1,i=1,…….,I --- (4)

ǫi≥0

The minimization of the objective function causes maximum separation between two classes with minimum number of points crossing their respective bounding planes. The parameter C is a regularization parameter that controls the trade-off between the two terms in the objective function. The proper choice of C is crucial for good generalization power of the classifier. The following decision rule is used to correctly predict the class of new instance with a minimum error. The advantage of the dual formulation is that it permits an efficient learning of non–linear SVM separators, by introducing kernel functions. Technically, a kernel function calculates a dot product between two vectors that have been (nonlinearly) mapped into a high dimensional feature space. Since there is no need to perform this mapping explicitly, the training is still

Ƒ(x)=sgn[WT X-Y] ----------- (5)

Feasible although the dimension of the real feature space can be very high or even infinite. The parameters are obtained by solving the following non linear SVM dual formulation (in Matrix form),

Minimize LD(U)=1/2 uTQu – et u --------- (6)

Dtu=0,0 ≤ Ce

where Q=DKD and K is kernel matrix. The kernel function K(AAT) may be polynomial or RBF (Radial Basis Function) is used to construct hyper plane in the feature space, which separates two classes linearly, by performing computations in the input space. The decision function in this nonlinear case is given by

Ƒ(x)=sgn[(K(x,xT) * u – y --------- (7)

where u, the Lagrangian multipliers.

When the number of classes is more than two, then the problem is called multiclass SVM. There are two types of approaches for multiclass SVM the first method is called indirect method, several binary SVM's are constructed and the classifier's output are combined for finding the final class. In the second method called direct method, a single optimization formulation is considered. The formulation of one of the direct methods called Crammer and Singer Method is

$$\text{Minimize } 1/2\sum_{k=1}^{N}(W_k^T)W_k^{\square} + C\Sigma_i{}^{1}{}_{=1}\varepsilon i \quad \text{--(8)}$$

Subject to the constraints

$$W_{wi}^{T}\varnothing(x_i) - W_k^{T}\varnothing(x_i) \geq e_k^{t} - \varepsilon i\ \forall\ k \neq k_i \quad \text{-- (9)}$$

where ki is the class to which the training data xi belong,

$$e_k^i = 1 - C_k^i \text{ ---------} \quad (10)$$

$$C_k^i = \begin{cases} 1 if\ ki = k \\ 0 if\ ki \neq k \end{cases} \text{ --------} \quad (11)$$

The decision function for a new input data xi is given by

$$\hat{d}_j = \arg max\{f_k(x_j)\} \text{ -----} \quad (12)$$

$$f_k(x_j) = w_k^T\varnothing(x_j) - \gamma_k \text{ -----} \quad (13)$$

3. Classification feed forward Artificial Neural Network

The data used for training and testing consist of feature vectors with 9 features each.The classification classes are cancerous cell and noncancerous cell. The features were chosen so that the types of normal cells does not have to be distinguished. The best classification result has been obtained by using Feed forward Artificial Neural Network. Mat lab Neural Network Toolbox has been used to train and to test the network. The best network had 10 hidden layer neurons. The cross-validation has been used for more reliable training and testing.

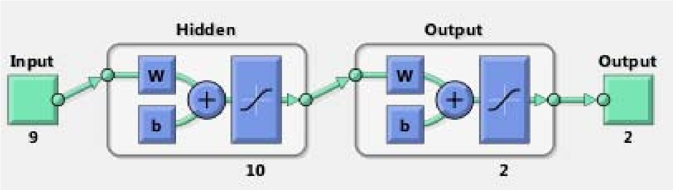

Figure 1: Simulink diagram for ANN

Neural networks consist of a large class of different architectures. In many cases, the issue is approximating a static nonlinear, mapping f (x) with a neural network f_{NN} (x), where X ∈ R^K .The most useful neural networks in function approximation are Multilayer Layer Perceptron (MLP) and Radial Basis Function (RBF) networks. Here we concentrate on MLP networks. The MLP consists of an input layer, several hidden layers, and an output layer. Node i, also called a neuron, in a MLP network is shown in Fig.2. It includes a summer and a nonlinear activation function g.

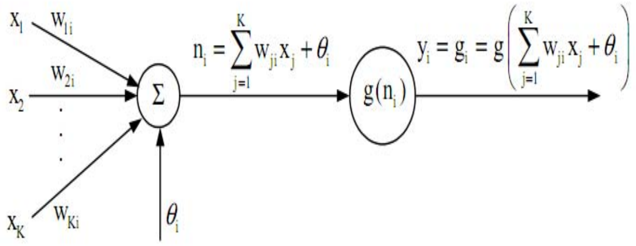
x_1
W_{1i}
x_2
W_{2i}
x_K
W_{Ki}
Σ
$n_i = \sum_{j=1}^{K} w_{ji} x_j + \theta_i$
$g(n_i)$
$y_i = g_i = g\left(\sum_{j=1}^{K} w_{ji} x_j + \theta_i\right)$
θ_i

Figure 2: Single node in a MLP network

The inputs *x*k, *k* = 1,...,*K* to the neuron are multiplied by weights *wki* and summed up together with the constant bias term $\theta_{I\,.}$The resulting n_i is the input to the activation function *g*. The activation function was originally chosen to be a relay function, but for mathematical convenience a hyberbolic tangent *(tanh)* or a sigmoid function are most commonly used. Hyberbolic tangent is defined as

$$\tanh(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad \text{-----} \quad (14)$$

The output of node *i* becomes

$$y_i = g_i = g\left(\sum_{j=1}^{K} w_{ji} x_j + \theta_i\right) \quad \text{---} \quad (15)$$

Connecting several nodes in parallel and series, a MLP network is formed. A typical network is shown in Fig. 3

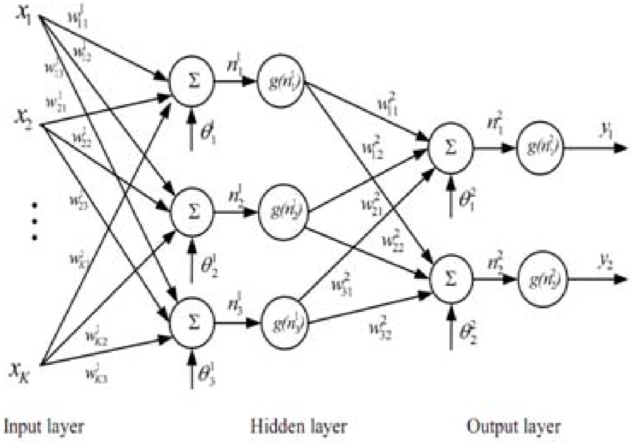

Figure 3: A multilayer perceptron network with one hidden layer. Here the same activation function g is used in both layers. The superscript of n , θ, or w refers to the layer, first or second.

The output *y*., *i* = 1,2, of the MLP network becomes

$$y_i = g\left(\sum_{j=1}^{3} w^2_{ji}\, g(n^1_j) + \theta^2_j\right) = g\left(\sum_{j=1}^{3} w^2_{ji}\, g\left(\sum_{k=1}^{K} w^1_{kj} x_k + \theta^1_j\right) + \theta^2_j\right) \quad \text{--} \ (16)$$

From (3) we can conclude that a MLP network is a nonlinear parameterized map from input space **x**∈ R^m to output space **y** ∈ $\mathbf{R}^m$ (here *m* = 3). The parameters are the weights $W^K{}_{JI}$and the biases $\theta\ ^{\mathbf{k}}{}_{j}$. Activation functions *g* are usually assumed to be the same in each layer and known in advance. In the figure the same activation function *g* is used in all layers.

Given input-output data (x, *y*.), *i* = 1,..., *N*, finding the best MLP network is formulated as a data fitting problem. The parameters to be determined are

$$\left(w^k_{ji}, \theta^k_j\right)$$

The procedure goes as follows.

First the designer has to fix the structure of the MLP network architecture: the number of hidden layers and neurons (nodes) in each layer. The activation functions for each layer are also chosen at this stage, that is, they are assumed to be known. The unknown parameters to be estimated are the weights and biases,

$$\left(\mathbf{w}^k_{ji}, \boldsymbol{\theta}^k_j\right)$$

Many algorithms exist for determining the network parameters. In neural network literature the algorithms are called learning or teaching algorithms, in system identification they belong to parameter estimation algorithms. The most well-known are back-propagation and Levenberg-Marquardt algorithms. Back-propagation is a gradient based algorithm, which has many variants. Levenberg-Marquardt is usually more efficient, but needs more computer memory. Here we will concentrate only on using the algorithms.

The parameters associated with the training algorithm like error goal, maximum number of epochs (iterations), etc, are defined. After the neural network has been determined, the result is first tested by simulating the output of the neural network with the measured input data. This is compared with the measured outputs. Final validation must be carried out with independent data.

Mat lab commands used in the procedure are *newff, train* and *sim.*

The mat lab command *newff* generates a MLPN neural network, which is

$$\boxed{\text{net= newff(}\ \underbrace{\text{PR}}_{\substack{\text{min,max}\\ \text{values}}}\ \text{,[}\underbrace{\text{S1 S2...SNl}}_{\text{size of the ith layer}}\text{],\{}\ \underbrace{\text{TF1 TF2...TFNl}}_{\text{activation function of ith layer}}\ \text{\}, }\underbrace{\text{BTF}}_{\substack{\text{training}\\ \text{algorithm}}}\text{)}}$$

The inputs in (4) are
R = Number of elements in input vector
xR= Rx2 matrix of min and max values for *R* input elements,
Si= Number of neurons (size) in the ith layer, *i = 1,...,Nl*
Nl = Number of layers *TFi* = Activation (or transfer function) of the *i*th layer, default = '*tansig*',
BTF = Network training function, default = *'trainlm'*
In Fig. 2 *R* = *K*, *S1*=3, *S2* = 2, *Nl* = 2 and *TFi* =g.

The default algorithm of command *newff* is Levenberg-Marquardt, *trainlm.* Default parameter values for the algorithms are assumed and are hidden from the user. They need not be adjusted in the first trials. Initial values of the parameters are automatically generated by the command. Observe that their generation is random and therefore the answer might be different if the algorithm is repeated.

After initializing the network, the network training is originated using *train* command. The resulting MLP network is called netl.

$$\text{net1} = \text{train}(\underbrace{\text{net}}_{\substack{\text{initial}\\\text{MLP}}}, \underbrace{x}_{\substack{\text{measured}\\\text{input vector}}}, \underbrace{y}_{\substack{\text{measured}\\\text{output vector}}})$$

The arguments are: *net*, the initial MLP network generated by *newff x*, measured input vector of dimension *K* and *y* measured output vector of dimension *m*.

To test how well the resulting MLP netl approximates the data, *sim* command is applied. The measured output is *y*. The output of the MLP network is simulated with *sim* command and called *ytest.*

$$\text{ytest} = \text{sim}(\underbrace{\text{net1}}_{\substack{\text{final}\\\text{MLP}}}, \underbrace{x}_{\substack{\text{input}\\\text{vector}}})$$

The measured output *y* can now be compared with the output of the MLP network *ytest* to see how good the result is by computing the error difference *e = y - ytest* at each measured point. The final validation must be done with independent data. In the following a number of examples are covered, where Mat lab Neural Network Toolbox is used to learn the parameters in the network, when input- output data is available.

3.1Training Function

1) trainbfg : Description trainbfg is a network training function that updates weight and bias values according to the BFGS quasi-Newton method.
2) traincgb : Description traincgb is a network training function that updates weight and bias values according to the conjugate gradient back propagation with Powell-Beale restarts.
3) Traingdx: Description traingdx is a network training function that updates weight and bias values according to gradient descent momentum and an adaptive learning rate.
4) Trainlm :Description trainlm is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization.

4. Experiment and Result

Breast Cancer classification algorithm would be carried out using Wisconsin Diagnosis Breast Cancer dataset created by Dr. William H. Wolberg at the University of Wisconsin. This dataset consists of 400 observations of patients with breast cancer among which 300 are benign and 100 are malignant status. Each instance has 20 features including id number and the class label that correspond to the type of breast cancer benign or malignant. These features are computed from digital image of fine needle of aspirates (FNA) of breast masses that describes the characteristics of the cell nuclei in the image.

The proposed work performed as two experiments. The first experiment is carried out by using SVM open source tool for multi class SVM, which uses Crammer and Singer Method. Second experiment aimed at assessing the effectiveness of and Neural Network using training function(trainbfg, traincgb, Traingdx, Trainlm). Discipulus is bundled with Notitia, which performs operations that include importing data from external sources, cleaning-up, and transforming and splitting data for use in Discipulus.

In both the experiments the training set consists of 80% of instances and testing set consists of 20% of instances of both benign and malignant classes. The performance of the SVM classifier is summarized in Table I.

Table 1: Classification Accuracy& Precision comparison(SVM)

Kernel function	*Accuracy*	*Precision*
mpl	86%	82%
Rbf	89%	85%
Guadratic	88%	88%
polynomial	88%	80%

Accuracy=(TP+TN)/(TP+TN+FP+FN) --- (17)

Precision= TP/(TP+FP) ------- (18)

Where TP and TN are True Positive and True Negative respectively, which are the pro-portion of positive and negative cases that were correctly identified. Positive cases are the records with Benign label and negative ones are with Malignant label. FP and FN stand for False Positive and False Negative which are the proportion of negative cases that were incorrectly classified as positive and the proportion of positive cases that were incorrectly classified as negative respectively [19]. Three types of Cross Validation (CV) technique were conducted in this paper (mpl, Guadratic, Polynomial and RBF kernel of SVM& trainbfg, traincgb, Traingdx, Trainlm of Neural Network). The classification accuracy& Precision resulted from each type was recorded, as shown in Fig.4.5

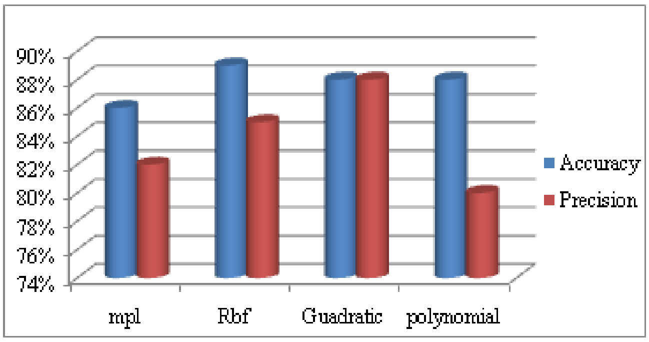

Figure 4: Classification Accuracy&Precision(Kernel function)

Table 2: Classification Accuracy & Precision comparison (Neural Network)

Traing function	Accuracy	Precision
trainbfg	88%	83%
traincgb	90%	85%
Traingdx	92%	88%
Trainlm	81%	71%

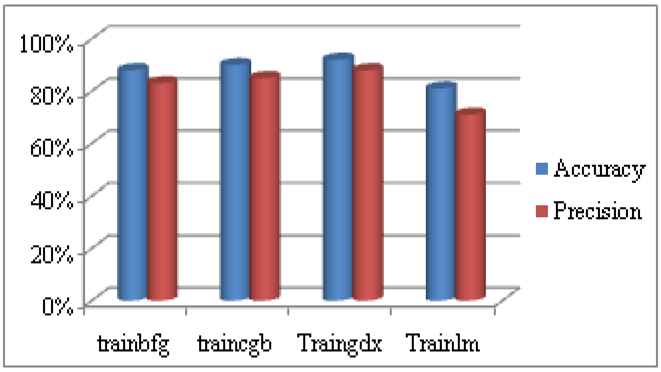

Figure 5: Classification Accuracy & Precision (Traing function)

5. Conclusion

This paper demonstrates the modeling of breast cancer as classification task and describes the implementation of Neural Network (NN) and Support Vector Machine (SVM) approach for classifying breast cancer as either benign or malignant. The results of both NN and SVM were compared on the basis of accuracy and precision. It was observed that classification implemented by Neural Network technique in this paper is more efficient compare to SVM as seen in the accuracy and precision. Based on the results, NN technique is more efficient compared to SVM technique in breast cancer detection.

References

[1] Sarvestan Soltani A, Safavi A A, Parandeh M N and Salehi M , "Predicting Breast Cancer Survivability using Data Mining Techniques",

[2] Software Technology and Engineering (ICSTE), 2nd International Conference, Vol.2, pages 227-231,2010.

[3] Werner J C and Fogarty T C, "Genetic Programming Applied to Severe Diseases Diagnosis", In Proceedings Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP), 2001.

[4] Iranpour M, Almassi S and Analoui M, "Breast Cancer Detection from fna using SVM and RBF Classifier", In 1st Joint Congress on Fuzzy and Intelligent Systems, 2007.

[5] Joachims T, Scholkopf B, Burges C and Smola A, "Making large-scale SVM Learning Practical, Advances in Kernel Methods-Support Vector Learning", Cambridge, MA, USA, 1999.

[6] Soman K P, Loganathan R and Ajay V, "Machine Learning with SVM and Other Kernel Methods", PHI, India, 2009.

[7] Crammmer Koby and Yoram Singer, "On the Algorithmic Implementation of Multi-class Kernel-based Vector Machines", Journal of Machine Learning Research, MIT Press, Cambridge, MA, USA, Vol.2, pages 265-292, 2001.

[8] Riccardo Poli, William B and Langdon Nicholas F McPhee, "A Field Guide to Genetic Programming", Lulu Enterprises, UK, 2008.

[9] Markus Brameier and Wolfgang Banzhaf, "A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining", IEEE Transactions on Evolutionary Computation, Vol.5, Issue No.1, 2001.

[10] Hussein A Abbass. An evolutionary artificial neural networks approach for breast cancer diagnosis, Artificial Intelligence in Medicine, 25(3):265{281, 2002.

[11] Pedro Henriques Abreu, Hugo Amaro, Daniel Castro Silva, Penousal Machado, and Miguel Henriques Abreu, Personalizing breast cancer patients with heterogeneous data, In The International Conference on Health Informatics, pages 39-42. Springer,2014.

[12] Mehmet Fatih Akay. Support vector machines combined with feature selection for breast cancer diagnosis. Expert systems with applications, 36(2):3240-3247, 2009.

[13] Mafaz Mohsin Al-Anezi, Marwah Jasim Mohammed, and Dhufr Sami Hammadi. Artificial immunity and features reduction for effective breast cancer diagnosis and prognosis. International Journal of Computer Science Issues (IJCSI), 10(3), 2013.

[14] Musrrat Ali, Millie Pant, and Ajith Abraham. Simplex differential evolution. Acta Polytechnica Hungarica, 6(5):95-115, 2009.

[15] Y Benchaib, Alexis Marcano-Cede~no, Santiago Torres-Alegre, and Diego Andina. Application of artificial metaplasticity neural networks to cardiac arrhythmias classification. In Natural and Artificial Models in Computation and Biology, pages 181-190. Springer, 2013.

[16] Cheng-Min Chao, Yao-Lung Kuo, and Bor-Wen Cheng. Three artificial intelligence techniques for finding the key factors in breast cancer, Journal of Statistics and Management Systems, 15(2-3):389-404, 2012.

[17] Hui-Ling Chen, Bo Yang, Gang Wang, Su-Jing Wang, Jie Liu, and Da-You Liu. Sup-port vector machine based diagnostic system for breast cancer using swarm intelligence. Journal of medical systems, 36(4):2505-2519, 2012.

[18] Kun-Huang Chen, Kung-Jeng Wang, Min-Lung Tsai, Kung-Min Wang, Angelia Melani Adrian, Wei-Chung Cheng, Tzu-Sen Yang, Nai-Chia Teng, Kuo-Pin Tan, and Ku-Shang Chang. Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm. BMC bioinformatics, 15(1):49, 2014.

[19] Smruti Rekha Das, Pradeepta Kumar Panigrahi, Kaberi Das, and Debahuti Mishra. Im-proving rbf kernel function of support vector machine using particle swarm optimization. International Journal, 2012.

[20] Kris De Brabanter, Peter Karsmakers, Jos De Brabanter, Johan AK Suykens, and Bart De Moor. Confidence bands for least squares support vector machine classifiers: A regression approach. Pattern Recognition, 45(6):2280- 2287, 2012.

[21] Mei-Ling Huang, Yung-Hsiang Hung, Wen-Ming Lee, RK Li, and Tzu-Hao Wang. Usage of case-based reasoning, neural network and adaptive neuro-fuzzy inference system classification techniques in breast cancer dataset classification diagnosis. Journal of medical systems, 36(2):407-414, 2012.

[22] Dervis Karaboga and Selfukokdem. A simple and global optimization algorithm for engineering problems: Differential evolution algorithm. Turkish Journal of Electrical Engineering & Computer Sciences, 12(1), 2004.

[23] R Karais, M Tez, YA Kilif, Y Kuru, and IGuler. A genetic algorithm model based on artificial neural

network for prediction of the axillary lymph node status in breastcancer. Engineering Applications of Artificial Intelligence, 26(3):945-950, 2013.

[24] CD Katsis, I Gkogkou, CA Papadopoulos, Y Goletsis, PV Boufounou, and G Stylios. Using artificial immune recognition systems in order to detect early breast cancer. International Journal of Intelligent Systems & Applications, 5(2), 2013.

[25] Javed Khan, Jun S Wei, Markus Ringner, Lao H Saal, Marc Ladanyi, Frank Wester-mann, Frank Berthold, Manfred Schwab, Cristina R Antonescu, Carsten Peterson, et al, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nature medicine, 7(6):673-679, 2001.

[26] Masayuki Kurosaki, Yasuhito Tanaka, Nao Nishida, Naoya Sakamoto, Nobuyuki Enomoto, Masao Honda, Masaya Sugiyama, Kentaro Matsuura, Fuminaka Sugauchi, Ya- suhiro Asahina, et al. Pre-treatment prediction of response to pegylated-interferon plus ribavirin for chronic