Computer–aided diagnosis of diabetes using least square support vector machine

Behnaz Naghash Almasi, Omid Naghash Almasi*, Mina Kavousi, Amirhossein Sharifinia

Department of Medical Science, Faculty of Nursing and Midwifery, Islamic Azad University, Mashhad, Iran
Department of Electrical Engineering, Islamic Azad University, Gonabad Branch, Iran
Department of Electrical Engineering, Semnan University, Semnan, Iran
Department of Electrical Engineering, Islamic Azad University, Mashhad Branch, Iran
*Corresponding author E-mail: o.almasi@ieee.org

Abstract

Diabetes incidence is one of the most serious health challenges in both industrial and developing countries; however, it is for sure that the early detection and accurate diagnosis of this disease can decrease the risk of affiliation to other relevant disease in diabetes patients. Because of the effective classification and high diagnostic capability, expert systems and machine learning techniques are now gaining popularity in this field. In this study, Least square support vector machine (LS-SVM) was used for diabetes diagnosis. The effectiveness of the LS-SVM is examined on Pima Indian diabetes dataset using k-fold cross validation method. Compared to thirteen well-known methods for the diabetes diagnosis in the literature, the study results showed the effectiveness of the proposed method.

Keywords: *Diabetes disease diagnosis, k-fold cross validation, Medical diagnosis, Least square support vector machine, Pima Indian Diabetes Datasets.*

1   Introduction

Diabetes is a major health problem in both developed and developing countries, and its morbidity is increasing. It is a disease in which the body unable to produce or properly use insulin [1]. It is proved that diabetes increases the risks of kidney disease, blindness, nerve damage, blood vessel damage and it contributes to heart disease.
The cause of diabetes continues to be a mystery, although both genetics and environmental factors such as obesity and lack of exercise appear to play roles [2]. The most common form of diabetes is Type 2 diabetes [3]. This type diabetes results from insulin resistance (a condition in which the body fails to properly use insulin), combined with relative insulin deficiency.
Although detection of diabetes is improving, about half of the patients with Type 2 diabetes are undiagnosed and the delay from disease onset to diagnosis may exceed 10 years. Thus, earlier detection of Type 2 diabetes and treatment of hyperglycemia and related metabolic abnormalities is of vital importance.
Fortunately, in recent years with an increased emphasis on diagnostic techniques and more effective treatments, the mortality rate from diabetes has declined. A key factor in this approach is the early detection and accurate diagnosis of this affliction [4-6].
Undoubtedly, the evaluation of data taken from patients and decisions of experts are the most important factors in diagnosis. Therefore, the use of classifier systems in medical diagnosis has been gradually increasing. After all, expert systems and various artificial intelligence techniques for classification also help experts to a considerable extent. Classification systems can help minimize possible errors that might occur due to in experienced experts, and also provide medical data to be examined in shorter time and more detailed [6, 7].
Proposed as effective statistical learning methods for classification [8], Support Vector Machines (SVMs) rely on support vectors (SV) to identify the decision boundaries between different classes. Nonlinearly related to the input space, SVM is based on a linear machine in a high dimensional feature space, which has allowed the development of somewhat quick training techniques, despite the large number of input variables and large training sets. SVMs have successfully been used to address many problems including handwritten digit recognition [9], object recognition [10], speaker identification [11], face detection in images [12], and text categorization [13].

The Least Square Support Vector Machine (LS-SVM) was first proposed by Suykens and et al. by modifying the formulation of standard SVM [14]. The LSSVM was modified at two points: First, instead of inequality constraints, it takes equality constraints and changed the quadratic programming to a linear programming. Second, a squared loss function is taken from the error variable [14, 15].

In this study, LS-SVM was employed to diagnose the diabetes disease. For training and testing experiments, Pima Indian diabetes dataset taken from the University of California at Irvine (UCI) machine learning repository was used. It was observed that the proposed method yielded the highest classification accuracies among the thirteen other methods in the literature. In this study, the performance was evaluated by the well-known k-fold cross validation method.

The rest of the paper is organized as follows. Section 2 briefly discusses the methods and results of previous studies on diabetes disease diagnosis. Section 3 reviews basic SVM and LS-SVM concepts, respectively. Section 4 elaborates on the Pima Indian diabetes dataset. Section 5 presents the experimental results achieved by applying the proposed method to diagnose diabetes. Finally, we would make our concluding remarks in section 6.

2 Review of literature

A great deal of approaches has been proposed to deal with automated diagnosis of diabetes with Pima Indian diabetes disease, and most of them have managed to achieve high generalization performances.

Deng et al. obtained 78.4% classification accuracy using ESOM [16]. Kayaer et al. [17] achieved 77.08% classification accuracy using multilayer neural network with LM algorithm. They have used conventional (one training and one test) validation method. Temurtas et al. use the probabilistic neural network (PNN) and they obtained 78.05% classification accuracy [18]. Smith et al. [19] proposed a neural network ADAP algorithm to build associative models and acheive to accuracy of 76%. Quinlan [20] applied C4.5 algorithm and the classification accuracy was 71.1%. Sahan et al. [21] used Attribute Weighted Artificial Immune System with 10-fold cross validation method and obtained a classification accuracy of 75.87%. Kumari and Chitra used SVM as famous computer aid diagnostic system in diabetes disease and obtained accuracy of 78% [22]. In [23], the researcher reached 76.73% accuracy with SSVM approach. Moreover, in order to prepare the performance of the LS-SVM in automated diagnostics, five different variant of Artificial Neural Networks (ANNs) were employed in this study, which are frequently used in the literature. There are different kinds of ANNs, which are determined by their training algorithms and topologies. Adjusting the weights and bias of the ANN, to train an ANN, means to select a model from the set of allowed models that minimize the error of the generalization criterion. In this study, three training algorithms were used for training a three-layer ANN. The first is a well-known Levenberg-Marquardt Back Propagation (LM BP), the second is Gradient Descent Back Propagation (GD BP), the third is Gradient Descent with Momentum Back Propagation (GDM BP), and the fourth is Gradient Descent with Adaptive learning rule Back Propagation (GDA BP). The fifth comparison method is Radial Basis Function (RBF), which turned out as a famous variant of ANNs.

3 Support vector machines for classification

In this section, we summarize the basic SVM concepts with regard to typical two-class classification problems. Support vector machines (SVM) originally developed by Boser et al. [24] and Vapnik [25], is based on the Vapnik–Chervonenkis (VC) theory and structural risk minimization (SRM) principle [25], by trying to find a trade-off between minimizing the training set error and maximizing the margin to achieve the best generalization ability and remain resistant to over fitting. Moreover, one major advantage of SVM is its use of convex quadratic programming, which provides only global minima; therefore, it avoids being trapped in local minima. For more details, cf. [25, 26], which give a complete description of the theory of SVM. In this section we will discuss the basic SVM concepts for typical binary-classification problems.

3.1 Linear separable case-hard margin SVM

Let us consider a binary classification task: $\{x_i, y_i\}, i = 1, \ldots, l,\ y_i \in \{-1,\ 1\}$, $x_i \in R_d$, where x_i are data points and y_i are corresponding labels. They are separated with a hyper plane given by $W^T x + b = 0$, where w is an n-dimensional coefficient vector which is normal to the hyperplane and b is the offset from the origin.

There are lots of hyperplanes that can separate the two classes, whereas the decision boundary should be as far away from the data of both classes as possible, the support vector algorithm seeks an optimal separating hyper plane that maximizes the separating margin between the two classes of data. As the wider margin can acquire the better

generalization ability, we can define a canonical hyper plane [25] such that $H_1 : W^T x^+ + b = +1$ for the closet points on one side and $H_2 : W^T x^- + b = -1$ for the closest on the other. Now to maximize the separating margin is equivalent to maximizing the distance between hyper plane H_1 and H_2. Hence we can get the maximal width between them: $m = (x^+ - x^-) . \frac{w}{\|w\|} = \frac{2}{\|w\|}$. To maximize the margin the task is therefore:

$$Min \, g(w) = \frac{1}{2} \|w\|^2$$
$$s.t.$$
$$y_i (w^T x_i + b) \geq 1, \forall i \qquad (1)$$

Therefore, the learning task could be reduced to minimization of the primal Lagrangian:

$$Min \, L_p (w,b,\alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - 1) \qquad (2)$$

where α_i are Lagrangian multipliers, hence $\alpha_i > 0$. The minimum with respect to b and w of the Lagrangian, L_p, is given by,

$$\frac{\partial L_p}{\partial b} = 0 \rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$
$$\frac{\partial L_p}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad (3)$$

Now we substitute back b and w in the primal, which gives the dual Lagrangian:

$$Max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_i x_i^T x_j$$
$$s.t.$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0 \qquad (4)$$

Obviously, it is a quadratic optimization problem (QP) with linear constraints. From Karush Kuhn–Tucker (KKT) condition, we know that: $\alpha_i (y_i (W^T X_i + b) - 1) = 0$, Thus, only support vectors have $\alpha_i \neq 0$, which carry all the relevant information about the classification problem. Hence the solution has the form: $W = \sum_{i=1}^{n} \alpha_i y_i x_i = \sum_{i \in SV} \alpha_i y_i x_i$, where SV is the number of support vectors. And gets b from $y_i (W^T X_i + b) - 1 = 0$, where x_i is support vector. Therefore, the linear discriminant function takes the form: $g(x) = W^T x + b = \sum_{i \in SV} \alpha_i y_i x_i^T x_i$.

3.2 Linear non-separable case-soft margin SVM

In practice, it is impossible to classify two classes accurately, because the data is always subject to noise or outliers, so in order to extend the support vector algorithms and solve imperfect separation, positive slack variables $\xi_i = 1, \ldots, l$ [25, 26] are introduced to allow misclassification of noisy data points, and to take into account the misclassification errors a penalty value C is introduced for the points that cross the boundaries. In fact, parameter C can be viewed as a way of controlling over-fitting. Therefore, the new optimization problem can be reformulated as follows:

$$Min g(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

s.t.

$$y_i (w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

(5)

Translate this problem into a Lagarangian dual problem

$$Max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_i x_i^T x_j$$

s.t.

$$0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0$$

(6)

The solution to this minimizations problem is identical to the separable case except for the upper bound C on the Lagrange multipliers α_i.

3.3 Non-linear separable case-kernel trick

In most cases, one can't linearly separate the two classes. In order to extend the linear learning machine to work well with nonlinear cases, a general idea is introduced, i.e., the original input space can be mapped into some higher-dimensional feature space where the training set is separable. With this mapping, the discriminant function is of following form:

$$g(x) = W^T \phi(x) + b = \sum_{i \in SV} \alpha_i \phi(x_i)^T \phi(x) + b$$

(7)

where $x_i^T x_i$ in the input space is represented as the form of $\varphi(x_i)^T \varphi(x)$ in the feature space. The functional form of the mapping $\varphi(x_i)$ does not need to be known since it is implicitly defined by the choice of kernel:

$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. Thus, the optimization problem can be rewritten as:

$$Max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

s.t.

$$0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0$$

(8)

After the optimal values of α_i have been found, the decision function would be based on the sign of:

$$g(x) = \sum_{i \in SV} \alpha_i y_i K(x_i, x) + b$$

(9)

As a rule, any positive semi-definite functions K(x, y) that satisfy Mercer's condition could be kernel functions [27]. Kernel function is defined as a function that corresponds to a dot product of two feature vectors in some expanded feature space. There are many kernel functions that can be employed in SVM. The most commonly used kernels in SVM are listed in Table 1.

Table 1: The conventional Kernel function

Name	Kernel Function expression
Linear Kernel	$k(x, x_i) = x^T x_i$
Polynomial Kernel	$k(x, x_i) = (t + x^T x_i)^d$
RBF Kernel	$k(x, x_i) = \exp(-\|x - x_i\|^2 / \sigma^2)$
MLP Kernel	$k(x, x_i) = \tanh(\beta_0 x^T x_i + \beta_1)$

In this Table σ and d are constants and those parameters must be set by a user. For MLP kernel a suitable choice for β_0 and β_1 is needed to enable the kernel function to meet Mercer's condition.

3.4 Least square support vector regression

The Least Square Support Vector Regression (LS-SVR) fully described in [28], is considered as an approximation tool in this study. The formulation of SVR was modified by Suykens and et al. at two points: First, instead of inequality constraints, it takes equality constraints and changed the quadratic programming to a linear programming. Second, a squared loss function is taken from the error variable [28, 29]. These modifications greatly simplified the problem and can be specifically described as follows:

$$\min J(w,e) = \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{k=1}^{N} e_k^2$$

$$s.t.$$

$$y_k = w^T \varphi(x_k) + b + e_k, \qquad k = 1,...,N$$

(10)

where e_k are error variables that play a similar role as the slack variables ξ_k in Vapnik SVM formulation and γ is a regularization parameter in determining the trade-off between minimizing the training errors and minimizing the model complexity.

The Lagrangian corresponding to (10) can be defined as:

$$L(w,b,e,\alpha) = J(w,e) - \sum_{K=1}^{N} \alpha_k \{w^T \varphi(x_k) + b + e_k - y_k\}$$

(11)

where $\alpha_k \in R$ are the Lagrange multipliers. The KKT optimality conditions for a solution can be obtained by partially differentiating with respect to w, b, e_k, and α_k

$$\begin{cases} \dfrac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{k=1}^{N} \alpha_k \varphi(x_k) \\[2mm] \dfrac{\partial L}{\partial b} = 0 \rightarrow \sum_{k=1}^{N} \alpha_k = 0 \\[2mm] \dfrac{\partial L}{\partial e_k} = 0 \rightarrow \alpha_k = \gamma e_k, k = 1,...,N \\[4mm] \dfrac{\partial L}{\partial \alpha_k} = 0 \rightarrow wT \varphi(x_k) + b + e_k - y_k = 0, \qquad k = 1,...,N \end{cases}$$

(12)

After elimination of the variable w and e_k, the following linear equation can be obtained:

$$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 & \vec{1}_N \\ \vec{1}_N & \Omega + \gamma^{-1}I_N \end{bmatrix} \begin{bmatrix} 0 \\ y \end{bmatrix}$$

(13)

where $y = [y_1,...,y_N]$, $\vec{1}_N = [1,...,1]$ and $y = [\alpha_1,...,\alpha_N]$. The kernel trick is applied here as follows

$$\Omega_{kl} = \varphi(x_k)^T \varphi(x_l) = K(x_k, x_l), \qquad k, l = 1, \ldots, N \tag{14}$$

where $K(.,.)$ is the kernel function meeting Mercer's condition. b and α can be obtained by the solution to the linear system

$$b = \frac{\vec{1}_N (\Omega + \gamma^{-1} I_N)^{-1} y}{\vec{1}_N^T (\Omega + \gamma^{-1} I_N)^{-1} \vec{1}_N^T} \tag{15}$$

$$\alpha = (\Omega + \gamma^{-1} I_N)^{-1} (y - \vec{1}_N^T b) \tag{16}$$

Eventually, the resulting LS-SVR model for function estimation can be expressed as:

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x_k, x_l) + b \tag{17}$$

3.5 Model selection

LS-SVMs have two adjustable sets of parameters. One of them is called kernel parameter(s) and the other is called regularization parameter (γ). LS-SVM generalization ability depends on the proper choosing of those parameters. The best performance of SVM is realized with an optimal choice of the kernel parameter(s) and the regularization parameter. The optimal choice of those parameters is called LS-SVM's model selection problem [30-32].

Kernel parameter(s) are implicitly characterizing the geometric structure of data in high dimensional space named feature space. In the feature space the data becomes linearly separable in such a way that the maximal margin of separation between two classes is reached. The selection of kernel parameter(s) will change the shape of the separating surface in input space. Selecting improperly large or small values in kernel parameter results Over-fitting or Under-fitting in the LS-SVM model surface, so the model would be unable to accurately separate data [33, 34].

In non-separable problems, noisy training data will introduce slack variables to measure their violation of the margin. Therefore, a penalty factor γ is considered for controlling the amount of margin violation. Other words, the penalty factor γ is defined to determine the trade-off between minimizing empirical error and structural risk error and also to guarantee the accuracy of classifier outcome in the presence of noisy training data. Higher γ values cause the margin to be hard and the cost of violation to become too high, so the separating model surface over-fits the training data. In contrast, lower γ values allow the margin to be soft, which results in under-fitting separating model surface. In both cases, the generalization performance of classifier is unsatisfactory, so it makes the LS-SVM model useless [33, 35].

In this research, we employ a grid-search technique [36] using 5-fold cross-validation to find out the optimal model selection of LS-SVM.

4 The Pima Indian diabetes disease diagnosis problem

In order to perform the research reported in this manuscript, Pima Indian diabetes dataset taken from the UCI machine learning respiratory were used [17-19, 37]. The reason for using this dataset is that because it is very commonly used among the other classification systems that we have used to compare this study with for Pima Indian diabetes diagnosis problem. All patients in this database are Pima Indian women at least 21 years old and living near Phoenix, Arizona, USA.

The dataset which consists of Pima Indian diabetes disease measurements contains two classes and 768 samples. The class distribution is

- Class 1: normal (500)
- Class 2: Pima Indian diabetes (268)

All samples have eight features. These features are:

1. Number of times pregnant (υ_1).

2. Plasma glucose concentration a 2 h in an oral glucose tolerance test (υ_2).

3. Diastolic blood pressure (mm Hg) (υ_3).

4. Triceps skin fold thickness (mm) (υ_4).

5. 2-h serum insulin (lU/ml) (υ_5).

6. Body mass index (weight in kg/(height in m)^2)   (υ_6).

7. Diabetes pedigree function Feature 8: 2-h serum insulin (lU/ml) (υ_7).

8. Age (years) (υ_8).

Although the dataset is labeled as there are no missing values, there were some liberally added zeros as missing values. Five patients had a glucose of 0, 28 had a diastolic blood pressure of 0, 11 more had a body mass index of 0, 192 others had skin fold thickness readings of 0, and 140 others had serum insulin levels of 0. After the deletion there were 460 cases with no missing values.

5   Experimental results and discussion

In this section, we introduce the performance evaluation method, which is used to evaluate the proposed method. Finally, we would present the experimental results and discuss our observations of the results. All the experiments reported here are implemented using RBF kernels for the following reasons:
When the relation between desired output and input attributes is nonlinear, the RBF kernel non-linearly maps datasets into the feature space so that it can handle the datasets. The number of hyper-parameters is the second reason which influences the complexity of model selection. The RBF kernel has less hyper-parameter than the polynomial kernel. Eventually, the RBF kernel is numerically less difficult [38-41].

5.1   Performance evaluation methods

In this study, *k*-fold cross validation method was used for performance evaluation of diabetes diagnosis using LS-SVM. *k*-fold cross validation is a way to improve over the holdout method. The data set is divided into *k* subsets, and the holdout method is repeated *k* times. Every time, one of the *k* subsets is used as the test set, and the other *k-1* subsets are gathered to form a training set. Then the average error across all *k* trials is calculated. The advantage of this method is that it is less significant for this method how the data gets divided. Every data point gets to be in a test set only once, and gets to be in a training set *k-1* times. As *k* increases, the variance of the resulting estimate reduces. The downside of this method is that the training algorithm must rerun *k* times from scratch, in other words, it takes *k* times computation to make an evaluation. To randomly divide the data into a test and training set *k* different times is a variant of this method. The advantage of this method is that you can independently choose how large you wish each test set to be and after how many trials you average should be over [42].

5.2   Results and discussion

We conducted some experiments on the Pima Indian diabetes disease dataset mentioned in section 4, so that we can evaluate the effectiveness of LS-SVM. We compared our results with those of earlier methods. Table 2 shows the classification accuracies of our method and ten previous methods. As the results show, our method using 10-fold cross validation has obtained the highest classification accuracy, 79.66% reported up to now.
Given the research findings, the SVM-based model that we have developed yielded very promising results in classifying the disease. We believe that the proposed system could be very helpful for physicians in their final decisions about their patients. Using such a tool, they can make reasonably accurate decisions.

6   Conclusion

Classification systems used in medical decision making, provide medical data to be examined in a shorter time and more detail. In this study, a medical decision making system based on LSSVM was applied in diagnosing diabetes and the most accurate learning methods were evaluated. To diagnose diabetes in a fully automatic manner using LSSVM,

Table 2: Classification accuracies obtained with LS-SVM and other classifiers from the literature

Method	Classification accuracy (%)
ESOM [16]	78.40
MLNN with LM [17]	77.08
PNN [18]	78.05
SVM [19]	75.53
C4.5 [20]	71.10
AWAIS [21]	75.87
SVM [22]	78.00
SSVM [23]	76.73
BP-GD	70.48
BP-GDM	71.36
BP-GDA	73.28
BP-LM	75.14
RBF	75.00
LS-SVM	**79.66**

experiments were conducted on the Pima Indian diabetes disease dataset. The experiment results strongly suggest that LSSVM could be helpful in diagnosis of disease. Compared to thirteen well-known methods in the literature, the experiment results demonstrated that the proposed method was more effective than other ten methods in the disease diagnosis.

References

[1] Ehab I. Mohamed, R. Linderm, G. Perriello, N. Di Daniele, S. J. Poppl, and A. De Lorenzo, "Predicting type 2 diabetes using an electronic nose-base artificial neural network analysis", Diabetes Nutrition & Metabolism, Vo.15, No. 4, 2002, pp.215−221.

[2] Kemal Polat, Salih Gunes, and Ahmet Aslan, "A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine", Expert Systems with Applications, Vol. 34, No.1, 2008, pp. 214−221.

[3] Rajendra Acharya, Peck Ha Tan, Tavintharan Subramaniam, Toshiyo Tamura, Kuang Chua Chua, Seach Chyr Ernest Goh, Choo Min Lim, Shu Yi Diana Goh, Kang Rui Conrad Chung, and Chelsea Law, "Automated identification of diabetic type 2 subjects with and without neuropathy using wavelet transform on pedobarograph", Journal of Medical Systems, Vol. 32, No.1, 2008, pp. 21−29.

[4] David West, Paul Mangiameli, Rohit Rampal, and Vivian West, "Ensemble strategies for a medical diagnosis decision support system: A breast cancer diagnosis application", European Journal of Operational Research, Vol. 162, No. 2, 2005, pp. 532–551.

[5] Kemal Polat and Salih Güneş, "Breast cancer diagnosis using least square support vector machine", Digital Signal Processing, Vol. 17, No.4, 2007, pp. 694–701.

[6] Hui-Ling Chen, Bo Yang, Jie Liu, and Da-You Liu, "A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis", Expert Systems with Applications, Vol. 38, No.7, 2011, pp. 9014–9022.

[7] Mehmet Fatih Akay, "Support vector machines combined with feature selection for breast cancer diagnosis", Expert systems with applications, Vol. 36, No.2, 2009, pp. 3240–3247.

[8] Vladimir N. Vapnik, "Statistical Learning Theory", New York: Wiley, 1998.

[9] Bernhard Scholkopf, Sung Kah-Kay, Christopher J. Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers", IEEE Transactions on Signal Processing, Vol. 45, No.11, 1997, pp. 2758–2765.

[10] MassimilianoPontil and Alessandro Verri, "Support vector machines for 3-D object recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol. 20, No. 6, 1998, pp. 637–646.

[11] Vincent Wan and William M. Campbell, "Support vector machines for speaker verification and identification", Proceedings of IEEE Workshop Neural Networks for Signal Processing, 2000, pp. 775–784.

[12] Edgar Osuna, Robert Freund, and Federico Girosi, "Training support vector machines: Application to face detection", In Proceedings of computer vision and pattern recognition, 1997, pp. 130–136.

[13] Thorsten Joachims, " Transductive inference for text classification using support vector machines", In Proceedings of international conference machine learning, vol. 99, 1999, pp. 200–209.

[14] Johan A. K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle, "Least squares support vector machines", World Scientific Publishing, 2002, Singapore.

[15] Johan A. K. Suykens, Joos Vandewalle, and Bart. De Moor, "Optimal control by least squares support vector machines", Neural Networks, Vol.14, No.1, 2001, pp. 23–35.

[16] Deng, D., & Kasabov, N. (2001). On-line pattern analysis by evolving self-organizing maps. In Proceedings of the fifth biannual conference on artificial neural networks and expert systems (ANNES) (pp. 46–51).

[17] Kayaer, K., & Yıldırım, T. (2003). Medical diagnosis on Pima Indian diabetes using general regression neural networks. In Proceedings of the international conference on artificial neural networks and neural information processing (ICANN/ICONIP) (pp. 181−184).

[18] Hasan Temurtas, Nejat Yumusak, Feyzullah Temurtas, "A comparative study on diabetes disease diagnosis using neural networks," Expert Systems with Applications, vol. 36, pp. 8610−8615, 2009.

[19] Smith, J.W., J. E. Everhart, et al.- "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus", Proceedings of the Symposium on Computer Applications and Medical Care (Washington, DC). R.A. Greenes. Los Angeles, CA, IEEE Computer Society Press, 1988, pp. 261-265.

[20] Quinlan, J.R. "C4.5: programs for machine learning", San Mateo, Calif., Morgan Kaufmann Publishers, 1993.

[21]   S.Sahan, K.Polat, H. Kodaz, and S. Gunes, "The medical applications of attribute weighted artificial immune system (awais): Diagnosis of heart and diabetes diseases", in ICARIS, 2005,p. 456-468.

[22]   V. Anuja Kumari, R.Chitra, "Classification Of Diabetes Disease Using Support Vector Machine", International Journal of Engineering Research and Applications, Vol. 3, No. 2, 2013, pp.1797-1801.

[23]   Santi Wulan Purnami, Abdullah Embong, Jasni Mohd Zain and S. P. Rahayu, "A New Smooth Support Vector Machine and Its Applications in Diabetes Disease Diagnosis", Journal of Computer Science, Vol. 5, No. 12, 2009, pp.1003-1008,.

[24]   Bernhard E. Boser, Isabelle M Guyon, and Vladimir N. Vapnik, "A training algorithm for optimal margin classifiers", In Fifth annual workshop on computational learning theory, 1992, pp. 144–152.

[25]   Corinna Cortes and Vladimir Vapnik, "Support-vector networks", Machine Learning, Vol. 20, No.3, 1995, pp. 273–297.

[26]   Nello Cristianini and John Shawe-Taylor, "An introduction to support vector machines: And other kernel-based learning methods", Cambridge, UK: Cambridge University Press, 2000.

[27]   Alexander J. Smola, "Learning with kernels: Support vector machines, regularization, optimization, and beyond", The MIT Press, 2002.

[28]   Johan A. K. Suykens, "Support vector machines: a nonlinear modeling and control perspective", European J. of Control, Vol. 7, No. 2–3, 2001, pp. 311–327.

[29]   Chen-Chia Chuang, "Fuzzy weighted support vector regression with a fuzzy partition," IEEE Transaction on System, Man, Cybern. B, Cybern., Vol. 37, No. 3, 2007, pp. 630–640.

[30]   Xinjun Peng and Yifei Wang, "A geometric method for model selection in support vector machine", Expert Systems with Applications, Vol. 36, No.3, 2009, pp. 5745–5749.

[31]   Shuzhou Wang, Bo Meng, "Parameter selection algorithm for support vector machine", Procedia Environmental Sciences, Vol. 11, 2011, pp. 538–544.

[32]   Olivier Chapelle, Vladimir N. Vapnik, Olivier Bousquet, and Sayan Mukherjee, "Choosing multiple parameters for support vector machines", Machine Learning, Vol. 46, No. 1, 2002, pp. 131–159.

[33]   S. Sathiya Keerthi, "Efficient Tuning of SVM Hyperparameters Using Radius/Margin Bound and Iterative Algorithms", IEEE Transaction on Neural Networks, Vol. 13, No. 5, pp.1225–1229.

[34]   Peter Williams, Sheng Li, Jianfeng Feng, and Si Wu, "A geometrical method to improve performance of the support vector machine", IEEE Transaction on Neural Networks, Vol. 18, No. 3, 2007,pp. 942–947.

[35]   Sheng Ding and Xiaoming Liu, "Evolutionary computing optimization for parameter determination and feature selection of support vector machines", IEEE Conference on Computational Intelligence and Software Engineering, 2009, pp. 1-5.

[36]   Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A practical guide to support vector classification", Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003. Available at http://www.csie.ntu.edu.tw/cjlin/libsvm/.

[37]   Christopher J. Merz and Patrick M. Murphy, "UCI repository of machine learning databases", http:www.ics.uci.edu:~mlearn:MLRepository.html, 1996.

[38]   S. Sathiya keerthi and Chih-Jen Lin, "Asymptotic behavior of support vector machines with Gaussion kenerl", Neural Computation, Vol. 15, No. 7, 2003, pp.1667–1689.

[39]   Hsuan-Tien Lin and Chih-Jen Lin, "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO –type methods", Technical report, Department of Computer Science , National Taiwan University, 2003, pp. 1–32.

[40]   Antoine Bordes, Seyda Ertekin , Jason Weston, and Leon Bottou ,"Fast Kernel Classifiers with Online and Active Learning", The Journal of Machine Learning Research, Vol. 6 , 2005, pp. 1579–1619.

[41]   Jiancheng Sun, Chongxun Zheng, Xiaohe Li, and Yatong Zhou, "Analysis of the Distance Between Two Classes for Tuning SVM Hyperparameters," IEEE Transaction on Neural Networks, Vol. 21, NO. 2, 2010, pp. 305–318.

[42]   Jeff Schneider's home page, http://www.cs.cmu.edu/~schneide/tut5/node42. html, last accessed August 2006.