International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:5, No:6, 2011

A Comparison of Different Soft Computing Models for Credit Scoring

Nnamdi I. Nwulu and Shola G. Oroja

***Abstract*—**It has become crucial over the years for nations to improve their credit scoring methods and techniques in light of the increasing volatility of the global economy. Statistical methods or tools have been the favoured means for this; however artificial intelligence or soft computing based techniques are becoming increasingly preferred due to their proficient and precise nature and relative simplicity. This work presents a comparison between Support Vector Machines and Artificial Neural Networks two popular soft computing models when applied to credit scoring. Amidst the different criteria's that can be used for comparisons; accuracy, computational complexity and processing times are the selected criteria used to evaluate both models. Furthermore the German credit scoring dataset which is a real world dataset is used to train and test both developed models. Experimental results obtained from our study suggest that although both soft computing models could be used with a high degree of accuracy, Artificial Neural Networks deliver better results than Support Vector Machines.

***Keywords*—**Artificial Neural Networks, Credit Scoring, Soft Computing Models, Support Vector Machines.

I. Introduction

THE credit score of an individual or group has overtime become the yardstick frequently used to determine their credit worthiness. Past records of the applicant is obtained and processed and a score is determined. A high or good credit score means that the applicant is suitable for business transactions while a low or bad credit score means that the applicant is not suitable for business transactions. There are four major classifications of credit scoring given in [3]. The first type is called application scoring and consists of an evaluation of the applicants demographic, social and other important information at the time of application. The aim is to determine if a new applicant qualifies to be given credit facilities. The second type of credit scoring is termed behavioural scoring which is quite similar to application scoring. The difference between both scoring methods is that while application scoring is for new or first time applicants, behavioural scoring is for old or returning applicants. Collection scoring is the third type of credit scoring and it categorises customers into various levels of insolvency or bankruptcy. This method of credit scoring basically distinguishes bankrupt customers determining those that require immediate action from those that don't. The fourth and final type of credit scoring is known as fraud detection which determines the degree to which an application is fraudulent.

Nnamdi I. Nwulu is with the Intelligent Systems Research Group, Near East University, Lefkosa, Mersin 10, Turkey (E-mail: ninwulu@ ieee.org).

Shola G. Oroja is with the Information Technology Department, Eastern Mediterranean University, Famagusta, Mersin 10, Turkey (e-mail: oroja.shola@yahoo.com).

Out of the four listed types of credit scoring, we consider only application scoring in this paper. This branch of credit scoring has hitherto been investigated in the literature with various results. A brief summary of prior works on credit scoring using soft computing approaches is provided below: In [1] the author designed different types of Artificial Neural Networks (ANN) for municipal credit rating classification for municipalities in the US. The ANN's designed are feed-forward neural networks (FFNN), Radial basis function neural networks (RBFNN), Probabilistic neural networks (PNN), Cascade correlation neural networks (CCNN), Group method of data handling (GMDH) polynomial neural networks and Support Vector Machines (SVM). Results obtained from this study show that the Probabilistic neural networks (PNN) happened to be the best performing neural network model as it provided the highest classification accuracy amongst the other tools. In another recent work [2] an SVM model was developed for predicting the degree of default in payments for technology based Small and Medium Enterprises (SME) in Korea. The model used input factors that consisted of company economic indicators, financial ratios and technology indicators. Results obtained in this work indicate that SVM outperforms neural networks and logistic regression models. In [3] the authors propose a missing data imputation method and subagging an ensemble classification technique for credit scoring using a real world dataset consisting of a sample of IBM's Italian customers. The authors apply several classifiers namely kernel support vector machines, nearest neighbours, decision trees and Adaboost and compare results with their corresponding subagged versions. Results obtained indicate that subagging significantly improves the classifiers performance and results.

In [4] authors utilized a generalized classification and regression tree (CART) for forecasting consumer credit risk for customer samples of a major commercial bank in the United States (US). The period studied is from January 2005 to April 2009 and the study determined that their machine learning model accurately forecasted credit events 3 to 12 months ahead.

It has been posited in recent research works that ensemble classification models or hybrid models often deliver better results for credit scoring.

In [5] a comparison is made of three ensemble methods namely Bagging, Boosting, and Stacking. The ensemble methods are built on Logistic Regression Analysis (LRA), Decision Tree (DT), Artificial Neural Network (ANN) and Support Vector Machine (SVM). In [6] a novel vertical bagging decision trees model (VBDTM) is proposed for credit scoring. The ensemble model is built on several base learners that consist of various types of machine learning single models, rule extraction models, two stages models, hybrid models and aggregation models and experimental results

indicate that the novel ensemble method (VBDTM) outperforms other methods. In [7] a host of several ensemble methods based on Least Square Support Vectors Machines (LSSVM) are developed and tested on two real word datasets. Examples of hybrid models include [8] where four different hybrid models were investigated. The models consisted of two classification techniques, two clustering techniques, a clustering technique used in conjunction with a classification technique and finally a classification technique used in conjunction with a clustering technique. All the developed hybrid models were evaluated using a real world dataset from a bank based in Taiwan and the obtained results from this experiment indicate that the two classification hybrid models (logistic regression and ANN's) outperformed the other hybrid models in terms of prediction accuracy. Another example of a hybrid approach is in [9] where the authors propose four hybrid SVM models for credit scoring. In each case SVM is combined with other tools and the resulting hybrid system is tested on two different datasets. The other tools used in combination with SVM include conventional statistical Linear Discriminate Analysis (LDA), Decision tree, Rough sets and F-score approaches. The results obtained from this experiment seem to indicate that LDA with SVM outperforms SVM alone.
It has been claimed that single models often have some inductive bias (SVM) and grapple with the problem of local minima and over-fitting (ANN), hence the need for ensemble classification techniques or hybrid models. Hybrid or ensemble models often times have a drawback being that they involve a fair amount of computational complexity which subsequently increases processing times (time costs) Moreover the evidence is inconclusive that they do obtain better results than single soft computing models.
There is no doubt however that single soft computing models with the right amount of care when pre processing data and designing model topology can deliver highly accurate judgments on credit scoring and in short processing times (fast processing). We therefore use two single and simple models: A Back Propagated Neural Network (BPNN) and a Radial Basis Function (RBF) Support Vector Machine (SVM) and perform a comparative analysis between these two soft computing single models for credit scoring. The criteria or yardstick for the comparative analysis is defined below and we determine amongst the two models:

- The model that returns a higher classification rate with minimal error.
- The model that obtained higher classification results with minimal time and computational costs
- The model that returned the highest accuracy with data it had not hitherto been exposed to (testing or validation data).

One of the aims of this paper is also to experiment with different normalization or scaling techniques and determine the best scaling/normalization technique which delivers higher accuracy rates. The remainder of this paper is arranged as follows; Section two describes the real world German credit scoring dataset. In section three we describe the steps taken in data pre-processing. In section four the architecture or topologies of the ANN and SVM learners are described whilst section five provides the results of implementing both soft computing schemes. Section six concludes the paper with suggestions for future works.

II. German Credit Scoring Dataset

The German credit scoring dataset is the dataset utilized in this paper. It is a real world dataset publicly available from the University of California online machine learning repository [10]. The dataset actually consists of details of real individuals with a mixture of both categorical and numerical attributes. There are 1000 customers in the dataset with 700 customers having a positive credit score which means it is advisable to do business with them and 300 with a negative credit score meaning business dealings are not advisable or is risky. In the original dataset there are 20 attributes (numerical and categorical). However we use an enhanced dataset available in the repository which has all the categorical variables transformed to numerical with the addition of four extra indicator variables bringing the total number of input attributes to 24. Every applicant has 24 responses to the attribute questions and a credit score of either accept/reject or positive/negative. Table I provides a summary of the dataset's input attributes, their types and the details excluding the four indicator variables.

TABLE I
Attributes In The German Credit Scoring Dataset

Attribute	Type	Detail
1	Categorical	Status of existing checking account
2	Numerical	Duration in months
3	Categorical	Credit History
4	Categorical	Purpose
5	Numerical	Credit account
6	Categorical	Savings account/bonds
7	Categorical	Present employment since
8	Numerical	Installment rate in percentage of disposable income
9	Categorical	Personal status and sex
10	Categorical	Other debtors/ guarantors
11	Numerical	Present residence since
12	Categorical	Property
13	Numerical	Age in years
14	Categorical	Other installment plans
15	Categorical	Housing
16	Numerical	Number of existing credits at this bank
17	Categorical	Job
18	Numerical	Number of people providing for
19	Categorical	Telephone
20	Categorical	Foreign worker

Prior to feeding the data into the soft computing models, data pre-processing is performed and in the next section we detail the different steps and procedure employed in pre processing data.

III. Data Pre- Processing

The German credit scoring dataset consists of a wide range of numerical values and thus cannot be fed into any of the soft computing models without being normalized / scaled. The usual normalization procedure involves determining the maximum numerical value in the entire dataset and then dividing all other numbers by that maximum numerical value. Using this approach on the German dataset is fraught with problems as the maximum value in this dataset is 184 which is far greater than the other numbers and dividing all the other numbers by this value will lead to most of the other input attributes being close to zero. This would make soft computing learning difficult, if not impossible and we believe efficient and careful normalization of the input data would greatly increase the accuracy of the developed soft computing models. In this paper three different normalization techniques are utilized and they are briefly described in the next sub sections.

A. Row or Horizontal Normalization

In this normalization technique, we move across the dataset horizontally and automatically determine the highest numerical value for each individual applicant or case, and divide all other numerical values by that maximum value. Thus we end up with 1000 maximum normalization values one for each applicant. Table II shows the maximum normalization values for the first 10 applicants.

TABLE II
Maximum Row Value For 10 Applicants Used For Normalizing Other Values in the Row

Applicant	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Max. Value	67	60	49	79	53	91	53	69	61	52

B. Column or Vertical Normalization

This normalization technique works by determining the highest numerical value for each of the 24 attributes and dividing all the other numerical values by that maximum value. Table III shows the maximum values for all the 24 attributes. These are the values all other input attributes are divided by.

TABLE III
Maximum Column Attribute Value for Dataset Used for Normalizing Other Input Values Column-wise

Attribute	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Max. Value	4	72	4	184	5	5	4	4	4	75	3	4
Attribute	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
Max. Value	2	2	2	1	1	1	1	1	1	1	1	1

C. SVM Normalization

This normalization technique is the normalization technique used for this dataset in LIBSVM [11]. Here all the values to the range of -1 to 1. Thus in this procedure unlike the prior scaling methods we have both positive and negative values in the final dataset.

IV. Training The Soft Computing Models

After data pre processing, the next procedure is training using the soft computing schemes. In this work Artificial Neural Networks (ANN) and Support Vector Machines (SVM) are the utilized soft computing schemes. The subsequent sub sections briefly describe the training or learning procedures employed.

A. Artificial Neural Network Learning Phase

For neural network training the learning algorithm used is the back propagation learning algorithm. Fig. 1 shows the architecture of the proposed credit scoring neural network model. Our ANN input layer has 24 neurons which corresponds to the number of the input attributes with each input neuron receiving a normalized attribute numerical value. There is one hidden layer containing 20 neurons while the output layer has 2 neurons where 1 0 represents a positive credit score and 0 1 denotes a negative credit score. In the original dataset 1 denotes a positive score while 2 denotes a negative score. We have simply recoded the 1 to 1 0 and the 2 to 0 1.

Since there were two methods of normalizing the neural network input data (row and column normalization) we end up with two neural networks where NN1 is the network for the row normalized data and NN2 is the network for column normalized data.

The initial weights of both neural networks were randomly generated with values between -0.35 and +0.35 and during training the learning co efficient and momentum rate were adjusted and the values that achieved the highest training accuracy rate were saved.

B. Support Vector Machines Learning Phase

The C-SVM model with an RBF kernel was used for SVM training using the LIBSVM software [11]. Furthermore in this work the cross validation approach was used. The cross validation procedure is used to avoid the problem of over fitting data. This approach allows us to determine suitable parameters (C and γ) for our RBF kernel. In v-fold cross-validation (which was used in this work), the training dataset is divided into v equal subsets. Sequentially one subset is tested using the SVM classifier trained on the remaining $(v - 1)$ subsets. The obtained cross-validation accuracy is the percentage of data correctly classified.

The highest cross validation accuracy is obtained and the parameters which produce the best results are saved and then used to train the SVM learner. The saved model is then used

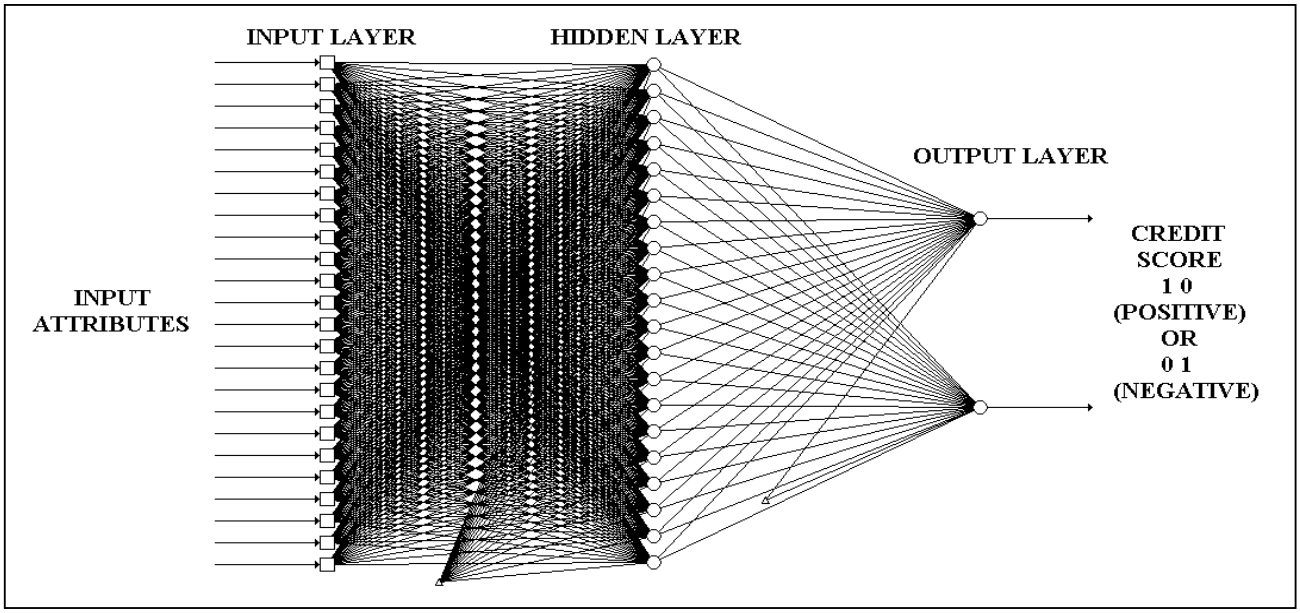

Fig. 1 Credit Scoring Artificial Neural Network Model Topology

on the out of sample data or data it has not been exposed to before (testing set). In this work v=5. The parameter search range for C was conducted from (2^{-100} – 2^{100}) while for γ it was also from (2^{-100} – 2^{100}). The best C obtained was 32 while γ was 0.0156. These values of C and γ were those that obtained the highest validation accuracy and they were then used for training the SVM learner.

V. Experimental Results and Discussions

The results of the soft computing credit scoring models were obtained using a 2.2 GHz PC with 2 GB of RAM, Windows XP OS and LIBSVM v 2.9.1 for the SVM model and MATLAB v 7.9.0 (R2009b) for the ANN model. Since there are a total of 1000 cases in the dataset we utilized training to testing ratio of 50%: 50%. Thus there are 500 cases for training the soft computing models and 500 cases for testing the trained models. The reason an equal training to testing ratio was used is to ensure that the soft computing models are not exposed to more training data than testing data thereby ensuring an unbiased model.

The following were the obtained results for the developed neural models: NN1 learnt and converged after 4000 iterations and within 452.03 seconds (training times), whilst the running time for the neural network after training and using one forward pass was 0.825×10^{-4} seconds (testing time). The training dataset accuracy rate was 87.66% while the testing dataset yielded an accuracy rate of 73.60%. Combining both accuracy rates (training and testing) yields an overall accuracy rate of 80.63%. For NN 2 error convergence was after 4000 iterations and within 452.78 seconds (training times), whilst the running time for the neural network after training and using one forward pass was 0.96×10^{-4} seconds (testing time). The training dataset accuracy rate was 92.09% while the testing dataset yielded an accuracy rate of 78.52%. Combining both accuracy rates (training and testing) yields an overall accuracy rate of 85.305%. Table IV lists the final parameters of the successfully trained neural network models and their accuracy rates and training times and Fig. 2 and Fig. 3 shows the error versus iteration graph for NN1 and NN2.

The trained SVM modelling system delivered the following results: using the training dataset yielded 84.4 % accuracy rate while the testing dataset yielded an accuracy rate of 83.6%. Combining both accuracy rates (training and testing) yields an overall accuracy rate of 84%. Table V lists the final parameters of the successfully trained SVM model.

TABLE IV
Final Parameters of the Credit Scoring Artificial Neural Network Learner

	NN1	NN2
Number of Input Neurons	24	24
Number of Hidden Neurons	20	20
Number of Output Neurons	2	2
Weights Values Range	-0.35 to +0.35	-0.35 to +0.35
Learning co-efficient	0.00084	0.00052
Momentum Rate	0.44	0.909
Obtained Error	0.0079	0.00062
Performed iterations	4000	4000
Training time(s)[1]	452.03	452.78
Run Time(s)[1]	0.825×10^{-4}	0.96×10^{-4}
Training dataset accuracy rate	87.66%	92.09%
Testing dataset accuracy rate	73.60%	78.52%
Overall accuracy rate	80.63%	85.305%

1. Using a 2.2 GHz PC with 2 GB of RAM, Windows XP OS and MATLAB v 7.9.0 (R2009b)

A comparison of all the obtained results along the criteria defined hitherto in this paper suggests that the ANN system achieves better accuracy rates while the SVM learner learnt in shorter time period. The SVM learning time does not take into consideration the time taken for parameter search which considerably increases SVM's training times and is not recorded here. It is our opinion however that the ANN modelling system would be more suitable for practical applications due to its obtained higher accuracy.

TABLE V
FINAL PARAMETERS OF THE CREDIT SCORING SUPPORT VECTOR MACHINES LEARNER

Number of Features	24
Number of Classes	2
C Parameter Search Range	$2^{-100} – 2^{100}$
γ Parameter Search Range	2^{-100} - 2^{100}
C	32
γ	0.0156
V	5
Type of SVM used	C-SVM
Kernel	RBF
Training optimization time(s)[1]	0.08
Training dataset accuracy rate	84.4%
Testing dataset accuracy rate	83.6%
Overall accuracy rate	84%

1. Using a 2.2 GHz PC with 2 GB of RAM, Windows XP OS and LIBSVM v 2.9.1.

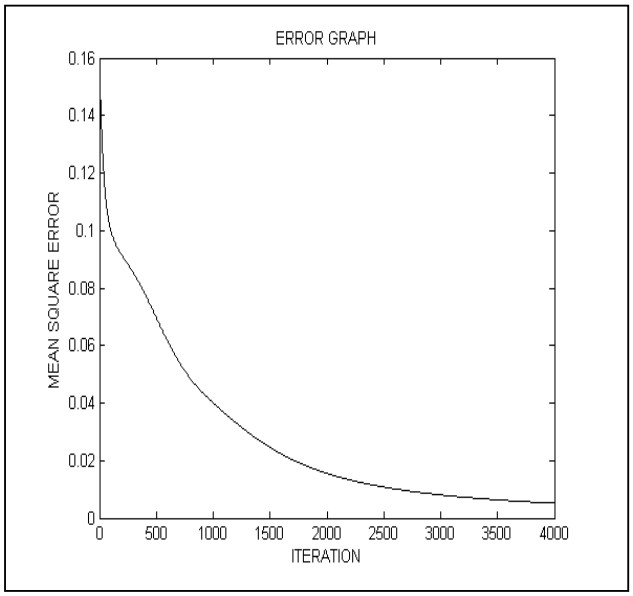

Fig. 2 Error Versus Iteration Graph for NN1

VI. CONCLUSION

In this paper, a comparison is provided between Support Vector Machines (SVM) and Artificial Neural Network (ANN) when applied to application credit scoring. The comparison between both soft computing models is along the lines of determining which of the models obtains a higher accuracy value with minimal computational complexity and processing times. The German credit scoring dataset publicly available online which contains 1000 examples of various credit applicants and their credit scores was used. This work has hitherto been attempted with various single soft computing models and various ensemble classification algorithms and hybrid systems with varying results.

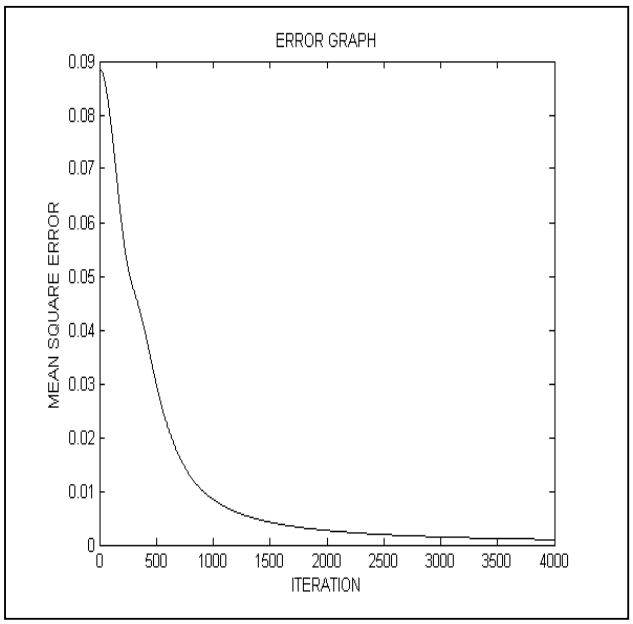

Fig. 3 Error Versus Iteration Graph for NN2

Before we feed the soft computing models with the data, data pre processing was performed. Data pre processing basically consists of normalizing and in this work three different ways of data normalization was investigated. Thus we ended up with three variations of the credit scoring dataset. Two of which were fed to two slightly similar ANN models while the last variation was fed to the SVM model.

All the ANN models in this work used the back propagation learning algorithm while the SVM learner used was the C-SVM with RBF kernel and *v*-fold cross validation mechanism. Experimental results obtained show that the better ANN system outperforms the SVM system (85.305% to 84%) while the SVM system requires a shorter training time than the ANN system (0.08 s to 452.03 s). The SVM training time does not however include the parameter search range which is a time consuming process and is not recorded here. It can therefore be concluded in light of the experimental results detailed above that the ANN system is the better system due to its higher accuracy rates and minimal time taken to train.

Any attempt to expand or improve this work would focus on training a soft computing model to determine when an applicant is on the borderline.

REFERENCES

[1] Hajek, P.: Municipal credit rating modelling by neural networks. *Decision Support Systems*. 51, 108--118 (2011)
[2] Hong,S.K., Sohn, S.Y.: Support vector machines for default prediction of SMEs based on technology credit. *European Journal of Operational Research*. 201, 838--846 (2010)
[3] Paleologo, G., Elisseeff, A., Antonini, G.: Subagging for Credit Scoring Models. *European Journal of Operational Research*. 201, 490--499 (2010)
[4] Khandani, A.E., Kim, A.J., Lo, A.W.: Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance* 34, 2767--2787 (2010)
[5] Wang, G., Hao, J., Ma,J., and Jiang, H.: A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications* 38, 223–230 (2011)
[6] Zhang,D., Zhou, X., Leung, S. C.H., Zheng, J.: Vertical bagging decision trees model for credit scoring. *Expert Systems with Applications* 37, 7838–7843(2010)
[7] Zhou, L., Lai, K.K. Yu,L.: Least squares support vector machines ensemble models for credit scoring. *Expert Systems with Applications* 37,127–133 (2010)
[8] Tsai, C.F., Chen, M.L.: Credit rating by hybrid machine learning techniques. *Applied Soft Computing* 10, 374--380 (2010)
[9] Chen, F.L., Li. F.C.: Combination of feature selection approaches with SVM in credit scoring. *Expert Systems with Applications* 37, 4902--4909 (2010)
[10] Asuncion, A., Newman, D.J.: UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science. (2007)
[11] Chang, C.C., Lin, C.J. : LIBSVM: a library for support vector machines. Software available at: /http://www.csie.ntu.edu.tw/cjlin/libsvm. (2001)