

Support Vector Machines Study on English Isolated-Word-Error Classification and Regression

¹Abu Bakar Hasan, ¹Tiong Sieh Kiong, ¹Johnny Koh Siaw Paw and ²Ahmad Kamal Zulkifl

¹College of Engineering,

²College of Foundation and General Studies, Universiti Tenaga Nasional, 43009 Kajang, Malaysia

Abstract: A better understanding on word classification and regression could lead to a better detection and correction technique. We used different features or attributes to represent a machine-printed English word and support vector machines is used to evaluate those features into two class types of word: *correct* and *wrong* word. Our proposed support vectors model classified the words by using fewer words during the training process because those training words are to be considered as personalized words. Those *wrong* words could be replaced by *correct* words predicted by the regression process. Our results are very encouraging when compared with neural networks, Hamming distance or minimum edit distance technique; with further improvement in sight.

Keywords: Artificial intelligence, communication, statistical theory, SVM kernel

INTRODUCTION

Support Vector Machines (SVM) are a useful technique for data classification (Vapnik, 1998; Chang and Lin, 2011) and regression (Vapnik, 1998; Smola and Scholkopf, 2003). SVM requires data to be presented as a vector of real numbers and scaling of data before applying SVM (Hsu *et al.*, 2010). In supervised learning problems, feature selection is important for generalization performance and running time requirements (Weston *et al.*, 2000; Dasgupta *et al.*, 2007), whereby a subset of the features available from the data are selected for application of a learning algorithm (Blum and Langley, 1997; Weston *et al.*, 2000). Selecting an optimal set of features is in general difficult, both theoretically and empirically (Kohavi and John, 1997).

In machine learning, satisfactory features selection is typically done by cross-validation, grid search and automatic scripts method (Hsu *et al.*, 2010; Chang and Lin, 2011), that is, thus in doing so it eliminates the irrelevant features or variables (Alain, 2003).

The tasks of error detection and correction must be clearly defined (Chang and Lin, 2011). Efficient techniques such as n-gram analysis, lookup tables, etc., have been devised for detecting strings that do not appear in a given word list, dictionary or lexicon (Kukich, 1992). Many existing spelling correctors use different techniques such as rule-based and minimum edit distance technique (Kukich, 1992). In this study, we are proposing Support Vector Classification (SVC) for detecting the types of words: *correct* or *wrong* and finally using Support Vector Regression (SVR) to

predict the possible correct words as the replacement for the wrong word.

This study has been organized in such a way that, in the next section, we provide brief literature survey on Support Vector Machines (SVM), methodology, experimental results and discussion and finally, conclusion.

THEORY ON SUPPORT VECTOR MACHINES

Basically, SVM is based on the Structural Risk Minimization principle (SRM) from the statistical learning theory (Vapnik, 1998). SVM is a set of related supervised machine learning methods used for classification, on the other hand, the empirical risk minimization principle, which is used by neural network to minimize the error on the training data, the SRM minimizes a bound on the testing error, thus allowing SVM to generalize better than conventional neural network (Gunn, 1997). Apart from the problem of poor generalization and overfitting in neural network, SVM also address the problem of efficiency of training and testing and the parameter optimization problems frequently encountered in neural network (Cristiani and Taylor, 2000). Support vectors in SVM can be categorized into two types: the training samples that exactly locate on the margin of the separation hyperplane and the training samples that locate beyond their corresponding margins, whereby the later is regarded as misclassified samples (Zhan and Shen, 2005). At the moment SVM usage generally covers data classification and regression only (Chang and Lin, 2011; Drucker *et al.*, 1996).

Given training data consisting of k labeled vectors represented by $\{x_i, y_i\}$ for $i = 1, 2, \dots, m$ where, $x_i \in R^n$ represents an n -dimensional input vector and $y_i \in \{-1, +1\}$ represents the class label. These training patterns are linearly separable if a vector w (orientation of a discriminating plane) and a scalar b (offset of the discriminating plane from origin) can be defined so that inequalities in Eq. (1) and (2) are satisfied (Nagi *et al.*, 2008):

$$w \bullet x_i + b \geq +1 \quad \text{if } y_i = +1 \quad (1)$$

$$w \bullet x_i + b \leq -1 \quad \text{if } y_i = -1 \quad (2)$$

A hyperplane which divides the data is to be determined. The procedure involve determining w and b so that for all values of y_i , Eq. (1) and (2) can be written as:

$$y_i(w \bullet x_i + b) \geq +1, \text{ for } i = 1, 2, \dots, m \quad (3)$$

If a hyperplane is optimally satisfied, then two classes are known to be linearly separable and Eq. (3) can be written as (Nagi *et al.*, 2008):

$$y_i(w \bullet x_i + b) = 0 \quad (4)$$

If the data is not linearly separable, another parameter called the slack variable ξ_i for $i = 1, 2, \dots, m$ is introduced with $\xi_i > 0$ such that Eq. (3) can be represented as (Nagi *et al.*, 2008):

$$y_i(w \bullet x_i + b) - 1 + \xi_i \geq 0 \quad (5)$$

The solution to find a generalized optimal separating hyperplane can be obtained using the condition (Nagi *et al.*, 2008):

$$\min \left[\frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \right] \quad (6)$$

The first term in Eq. (6) controls the learning capacity, while the second term controls the number of misclassified points. The parameter C is selected by the user (Nagi *et al.*, 2008), which is viewed as a regularization parameter that characterizes willingness to accept possible misclassifications in linearly non-separable datasets.

The classification function is (Nagi *et al.*, 2008):

$$\begin{aligned} f(x) &= \text{SIGN} \left[\sum_{i=1}^m y_i \alpha_i (w_i \bullet x_i + b) \right] \\ &= \text{SIGN} \left[\sum_{i=1}^m y_i \alpha_i K(x_i, x_j) + b \right] \end{aligned} \quad (7)$$

where, α_i is the support vector of nonnegative Lagrange multipliers. The parameters α_i and b are to be determined by SVM's learning algorithm (Aida, 2009) during the classification process. Parameters y_i and $K(x_i, x_j)$ are the label or class type and the Kernel function respectively. Parameter x_i is the training or test data while x_j is the input data for prediction.

On the other hand, the regression function is:

$$\begin{aligned} g(x) &= \sum_{i=1}^m (\alpha_i - \alpha_i^*) (w \bullet x_i + b) \\ &= \sum_{i=1}^m (\alpha_i - \alpha_i^*) K(x_i, x_j) + b \end{aligned} \quad (8)$$

For our case study, we have chosen the Radial Basis Function (RBF) as kernel as it has less numerical difficulties (Vapnik, 1998) and has the property defined by $K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2)$. The kernel bandwidth parameter γ and the penalty parameter C for the RBF kernel are pre-determined by v -fold cross validation or grid-search (Hsu *et al.*, 2010).

A recent result (Keerthi and Lin, 2003) shows that if RBF is used with model selection, then there is no need to consider the linear kernel.

METHODOLOGY ON SVM CLASSIFICATION AND REGRESSION MODEL

We defined *word* literally as a word that could be correctly read or identified without any ambiguity and non-word is defined as vice-versa. The scope of our study comes under the isolated-word error category (Kukich, 1992) and specifically on machine printed English word only. In this study, we are proposing a model to represent a word to meet the SVM classification requirements, with a maximum of eleven attributes or features namely numbers of consonant or non-vowel (nv) and vowel (v) character in them, total letter length (len) and their weight (w), weight of first (wI) and last letter (wN), position-vowel factor (pvf), sum of position-consonant-vowel factor (pcf), position factor (pf), real value (RV) and lastly, the type for the first letter (typ) in the word. In order to classify these words correctly, hereby we are proposing five rules as stated below:

Rule 1: The number of non-vowel and vowel in a word equals to the length of the word.

- Rule 2:** The weight of each word is equal to the word own weight. We define each character has a specific weight: (A, a) = 1, (B, b) = 2, (C, c) = 3, (D, d) = 4, until (Z, z) = 26 and (') = 0.
- Rule 3:** The type for first character of the word is set to 0 if the first character is a non-vowel and 1 if vowel.
- Rule 4:** The first letter in a word does not necessarily carry the highest probability that determine the word classification.
- Rule 5:** If the weight of the first letter in the first word is different from the first letter of the second word and consecutively, then either word is different.

To illustrate our proposed model, we take any word, for example, consider the word *final*, then, with the help of Rule 1, 2 and 3, it can be shown that $nv = 3$, $v = 2$, $len = 5$, $w = 42$, $wl = 6$, $wn = 12$, $pvf = 22$, $pcf = 108$, $pf = 130$, $typ = 0$ and $RV = 0.553132238$. We used LibSVM (Hsu *et al.*, 2010; Chang and Lin, 2011), which is a SVM learning tool package, for the training and testing process of the normalized words. The five rules above will mathematically play an important function in the classification and regression of each word.

Word classification: Proposing a 2-steps procedure as follows:

- Step 1:** Select the cross-validation parameter (ν) equal 10 and the kernel function (RBF). By choosing $\nu = 10$, LibSVM package (Chang and Lin, 2011) will automatically divides the training data into ten different parts with basically equal numbers of data. Train the ten parts of these training data to determine the best value for parameters C and γ . The best values refer to the highest efficiency percentage in classifying the training data. After the best (C , γ) was chosen, the whole training set is trained again to generate the classifier or classification model, in which support vectors from the trained data are being determined automatically.
- Step 2:** Test the unknown data to obtain the final classification accuracy which represents how good the model obtained in step 1 in classifying these data or words into two separate classes correctly: *correct* and *wrong* word.

Wrong word regression: Again, we are proposing 2-steps procedure:

- Step 1:** Choose RBF. Train the training data by setting the correct value for parameter C and γ obtained through cross-validation process.
- Step 2:** Test the same unknown to obtain the mean squared error and squared correlation coefficient, hence deciding the possible correct words.

EXPERIMENTAL RESULTS AND DISCUSSION

Word classification: We found that SVM can be trained on words with and without spelling errors, thus it has the potential to adapt to the unique patterns for each different word. This unique pattern characterized the personnel preference of a person to certain words. Meaning that a person tends to be personalized when comes to using or choosing a word; another person may prefer to use a different set of words during conversation, texting or writing. We define these preferences as the personalized dictionary of a particular person; thus, we managed to create a data base with a smaller number of words for SVM training purpose. To find the extent of our proposed classification model, we carried out two different studies in which all our data (training and testing) has been normalized accordingly to the range of (0, 1) and each word appears only once in our data base.

For the first work, we took three drafts of three different technical reports of a student, namely Report 1, Report 2 and Report 3, whereby normally these reports (Vivekan, 2010) have to be checked, corrected and commented by the respective supervisor before the reports are being allowed to be submitted to the University. We found that in those reports there are very few words with spelling errors made by the student himself. The pattern on the words used in writing three different reports are distinguishable for the student, meaning that he tends to use certain particular words in his writing. Since the total numbers of spelling errors in the reports are not many, thus, we have created 418 (or 26.2%) spelling errors ourselves from the total 1595 words. We decided to divide those data into 70:30 ratios, whereby 478 (30%) of the data will be used for testing purposes with $\nu = 10$. This time we obtained the testing accuracy of 77% with the best values of (C , γ) = (1, 1) by using iteration technique and number of attributes equals seven. In the second work, we increased the number of attributes and training words. Using an automatic script (Hsu *et al.*, 2010)

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\abakarh>cd desktop
C:\Documents and Settings\abakarh\Desktop>cd python
C:\Documents and Settings\abakarh\Desktop\python>easy.py train.txt test.txt
Scaling training data...
Warning: original #nonzeros 12200
         new       #nonzeros 12217
Use -l 0 if many original feature values are zeros
Cross validation...
Best c=0.03125, g=0.0078125 CV rate=69.5881
Training...
Output model: train.txt.model
Scaling testing data...
Testing...
Accuracy = 78.2705% (353/451) (classification)
Output prediction: test.txt.predict

C:\Documents and Settings\abakarh\Desktop\python>_
    
```

Fig. 1: Accuracy obtained using an automatic script

Table 1: Test results

No. of attributes used	No. of training word	No. of test word	Test accuracy (%)
7	1117	478	77
11	1142	451	78.27

(Fig. 1), subsequently we obtained the best parameters for (C, γ) as (0.03125, 0.0078125) respectively, thus giving a much better test accuracy value of 78.27% (Table 1). From the classification model file, we observed that parameter $\rho = 1.00380$ and number of support vectors is 695. Lastly, Kukich (1992) did mention that there is no optimal value for the number of attributes that one could consider the most appropriate to use.

In order to improve the accuracy, we have tried to rescale our data from (0, 1) range to (-1, 1) as proposed by Hsu *et al.* (2010). We observed no change in the value of the test accuracy for both data samples; meaning that our normalization of all the words to (0, 1) range is acceptable. For future work on the accuracy improvement possibility, one that we could look into is word pronunciation. Presently, our proposed model does not involve voice for pronunciation purpose as one of the attributes, meaning that a unique technique has to be deployed so that words can be ‘logically’ pronounced for the purpose of SVM processes.

Comparing work by Lyon and Yaeger (1996), Al-Jawfi (2009) and Khawaja *et al.* (2006) in classification of hand-printing English, Arabic and Chinese

characters using neural network respectively, we found that the selected attributes were mainly based on the character size or the pixel counts obtained from character own images. Rani *et al.* (2011) used Gabor filters to get the concentration of energies in various directions for identifying printed English numerals at word level from Punjabi words document and she has chosen 5-fold cross-validation. Our proposed SVM model is using maximum eleven different behavioral features in representing an English word and 10-fold Cross-Validation (CV).

Referring to Table 2, we observed that our study produced the highest accuracy; meaning that SVM could provide us the best tool to classify isolated-word error in texts. Table 3 shows the comparison with Frank and Asuncion (2010) on related work on British English and Telugu vowels and again the accuracy values obtained are reasonably acceptable due to the constraints in obtaining a large numbers of attributes.

During the classification process, one of the file produced was a model file from which we obtained the parameter $b = -\rho = 1.00380$, $\gamma = \text{gamma} = 0.0078125$ and the summation of Eq. (7) can be partly shown as in Table 4 for a set of prediction data $X_{j1} = 0.148260212$, $X_{j2} = 0.211119459$, $X_{j3} = 0.189854345$, $X_{j4} = 0.444444444$, $X_{j5} = 0.5$, $X_{j6} = 0.4375$, $X_{j7} = 0.363636364$, $X_{j8} = 0.48$, $X_{j9} = 0.56$, $X_{j10} = 0$ and $X_{j11} = 0.161035843$. We define parameter m in Eq. (7) to be the number of support vectors, that is, $m = i = \text{total number of SV} = 695$. Obviously, The RBF kernel is

Table 2: Comparison with Kukich (1992) study

Technique	1142-word lexicon test accuracy (%)
Minimum edit distance (use grope)	62.00
Similarity key (bocast token reconstruction)	78.00
Simple N-gram vector distance (use hamming distance)	68.00
Probabilistic (use kernighan-church-gale error probe)	78.00
Neural network (use back propagation classifier)	75.00
Support vector machines (use radial basis function)-our proposal	78.27

Table 3: Related comparison with Frank and Asuncion (2010)

Techniques	CV	Test accuracy (%)
SVM (use gauss C = 1, s = 0.1)	10x	76.6
Fuzzy (use SOM)	2x	73.5
SVM (use RBF)	N/A	53.5

Table 4: Partial calculation for kernel function $K(x_i, x_j)$ for $i = 1$ and 2

Iteration i	$\alpha_i y_i$	$f(x_i) = \alpha_i y_i$ Kernel $K(x_i, x_j) = \alpha_i y_i \exp(-\gamma x_i - x_j ^2)$
1	0.03125	0.030191835
2	0.03125	0.029991326

Table 5: Selected classification results

Original label y_i	Word	$\sum \alpha_i y_i \exp(-\gamma x_i - x_j ^2) + b$	$f(x) =$ Class	Comment
1	Hazardous	+1.004043878	+1	Classified correctly
-1	Hazarduos	-1.035413453	-1	Classified correctly
-1	Hazardous	-1.035413452	-1	Classified correctly
-1	Azadous	-1.004521435	-1	Classified correctly

Table 6: Selected regression results

Parameter (C, γ)	No. of SV	ρ or b	MSE	SCC
(1.0, 0.0078125)	754	0.927257	0.183823	0.00075181
(0.5, 0.0078125)	723	0.930138	0.183797	0.01334780
(0.25, 0.0078125)	716	0.918381	0.183816	0.00022885
(0.125, 0.0078125)	700	0.912727	0.183814	0.00630337

actually calculating the exponential of the negative of Euclidean distance between the support vectors and the unknown vector points that we want to predict.

Accordingly, it can be shown that for $i = 695$ is $f(x_{695}) = -0.029704383$. Hence, we can calculate the decision function when the offset parameter $b = 1.00435$ as:

$$f(x) = \text{SIGN} \left[\sum_{i=1}^{695} y_i \alpha_i K(x_i, x_j) + b \right] = \text{SIGN} [f(x_1) + f(x_2) + \dots + f(x_{695}) + b]$$

$$= \text{SIGN} [0.030191835 + 0.029991326 + \dots + (-0.029704383) + 1.00435] = \text{SIGN} [1.000000821] = +1 \quad (9)$$

Graphically $f(x)$ is a mapped point on the positive side of the hyperplane of the *feature space*, meaning that this test data has been classified into a+1 class. Rechecking with our data base shows that the test or predicted data used in the above calculated was the word 'maintain'. The word was originally labeled as Label 1 word which means it was a correctly printed English word. Table 5 shows the classification results on three other test data or words.

Wrong word regression: Table 6 shows that the regression results on our test data with different combination values of C and γ and the best results came from the best values of C and γ that were obtained by using iteration technique Hsu *et al.* (2010). It was observed that when (C, γ) are $(0.5, 0.0078125)$ respectively, we get the third smallest total number of support vectors which will provides a shorter computer processing time for regression, the largest distance between the regression's hyperplane from the origin, the smallest Mean Squared Error (MSE) and the best Squared Correlation Coefficient (SCC). Statistically, MSE measures the average of the squares of the "errors" where the error is the amount by which the value implied by the model differs from test data. The difference occurs because of randomness or because the model does not account for information that could produce a more accurate estimate (Wikipedia, 2012). On the other hand, the SCC is a statistical measure of how well the regression line approximates the real data points. An SCC of 1.0 indicates that the regression line perfectly fits the data (Wikipedia, 2012). What we observed was that the obtained SCC values are very low, even though the MSE values can be considered as good number (Table 6). Thus, we can safely say that our regression model might be facing some difficulties in predicting a correct word.

On the other hand, regression function $g(x)$ for each of the word (Table 7) indicates that the regression model or kernel that we obtained while training the train data was a good model, that is, $g(x)$ is always greater than 0.9. What we want from our study on regression is if someone typed any *wrong* word or words with label -1 such as *hazardous*, *hazarduos*, *azardous*, etc., the error (s) needs to be corrected either by the user herself or by a word corrector. Thus, in order to correct the error (s), we have calculated their Margin of Error (ME) which equals to ± 1.96 of the sample standard error (Wikipedia, 2012) for 95% confidence level with the number of sample of 451. We

Table 7: Regression function vs SVR parameters

Label	Test data or word	$g(x)$	NRV (with ME = 0.010038292 or ~1%)	Npvf (ME = 0.012966775 or ~1.3%)	Npcf (ME = 0.013435315 or ~1.3%)
-1	<i>Hazarduos</i>	0.911210143	0.239992837	0.413010590	0.278737791
-1	<i>Hazardous</i>	0.908146049	0.196463611	0.367624810	0.193839219
-1	<i>Azardous</i>	0.920249918	0.434020549	0.364599092	0.222389181
1	<i>Hazardous</i>	0.911334135	0.241136019	0.422087746	0.278737791
1	<i>Maintain</i>	0.902750076	0.161035843	0.148260212	0.211119459
1	<i>Mr</i>	0.908906607	0.089688493	0.000000000	0.036814425
1	<i>While</i>	0.912554745	0.126606160	0.078668684	0.065364388
1	<i>Non</i>	0.905171322	0.075795475	0.045385779	0.042073629
1	<i>Recommendation</i>	0.919826659	0.504113626	0.632375189	0.551465064
1	<i>Characterization</i>	0.921387587	0.666758859	0.757942511	0.887302780
1	<i>Comprehensive</i>	0.917891526	0.364321252	0.399394856	0.625093914

are proposing using only the three most important out of eleven parameters initially considered for our regression model, that is, parameter NRV, Npvf and Npcf. What we found was that the word *hazarduos* can be corrected and replaced by the word *hazardous* but not *hazardous* and *azardous*. These arguments are based on their calculated NRV, Npvf and Npcf values as shown in Table 7. Hence, the statistical method that we have adopted here has not been able to produce the expected results; even though the three respective parameter's margin of error is less than 1.4% only each. Hence, Table 7 contents need further analysis.

CONCLUSION

In this study, we have demonstrated the ability of SVM in classifying and regression certain isolated-word errors in the English word. One finding was that SVC could be trained as a word classification system, but for regression our SVR outputs need further fine tuning.

REFERENCES

Aida, Z., 2009. Development of an Intelligent Fault Detection System for Medium Voltage Circuit Breaker Compartment, M. Eng. Thesis, Universiti Tenaga Nasional, Malaysia.

Alain, R., 2003. Variable selection using SVM-based criteria, J. of Mach. Learn. Res., 3: 1357-1370.

Al-Jawfi, R., 2009. Handwriting Arabic character recognition LeNet using neural network. Int. Arab J. Inf. Technol., 6(3): 304-309.

Blum, A.L. and P. Langley, 1997. Selection of relevant features and examples in machine learning. Artif. Int., 97: 245-271.

Chang, C. C. and C. J. Lin, 2011. LIBSVM-A library for support vector machines. ACM Trans. Intell. Syst. Technol., 2(3):1-27.

Cristiani, N. and J.S. Taylor, 2000. An Introduction to SVM and Other Kernel-based Learning Methods, Cambridge University Press, UK.

Dasgupta, A., P. Drineas, B. Harb, V. Josifovski and M.W. Mahoney, 2007. Feature selection methods for text classification, ACM, pp: 230-239.

Drucker, H., C.J.C. Burges, L. Kaufman, A. Smola and V. Vapnik, 1996. Support Vector Regression Machines. Advances in Neural Information Processing Systems 9, NIPS 1996, MIT Press, pp: 155-161.

Frank, A. and A. Asuncion, 2010. UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, Retrieved from: [http:// archive. ics. uci. edu/ml](http://archive.ics.uci.edu/ml).

Gunn, S., 1997. SVM for Classification and Regression, Technical Report, Speech and Intelligent Systems Research Group, University of Southampton, USA.

Hsu, C.W., C.C. Chang and C.J. Lin, 2010. A practical guide to support vector classification. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, April 15.

Keerthi, S.S. and C.J. Lin, 2003. Asymptotic behavior of SVM with Gaussian kernel. Neural Comput., 15: 1667-1689.

Khawaja, A., A.F. Chando, A. Rajpar and A.R. Jafri, 2006. Classification of printed Chinese characters by using neural network, in Proc. of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, pp: 30-35.

Kohavi, R. and G.H. John, 1997. Wrappers for feature subset selection, Artif. Int., 97: 2.

Kukich, K., 1992. Techniques of automatically correcting words in text. ACM Comput. Surv., 24(4): 73-324.

Lyon, R.F. and L.S. Yaeger, 1996. On-line hand-printing recognition with neural networks. 5th Internal Conference on Microelectronics for Neural Networks and Fuzzy Systems, Lausanne, Switzerland, Feb. 12-14.

- Nagi, J., A.M. Mohammad, K.S. Yap, S.K. Tiong and S.K. Ahmed, 2008. Non-technical loss analysis for detection of electricity theft using support vector machines, Proceeding of Second IEEE International Conference on Power and Energy (PECON), Dec. 1-3, pp: 907-912.
- Rani, R., R. Dhir and G.S. Lehal, 2011. Identification of printed punjabi words and english numerals using gabor features. *World Acad. Sci. Eng. Technol.*, 73: 392-395.
- Smola, A.J. and B. Scholkopf, 2003. A Tutorial on Support Vector Regression. *Neuro COLT Technical Report TR-98-030*, pp: 1-19.
- Vapnik, V.N., 1998. *Statistical Learning Theory*, Wiley, New York.
- Vivekan, M., 2010. Internal Technical Reports, UNITEN, Malaysia.
- Weston, J., S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik, 2000. Feature selection for SVM. *Adv. Neur. Inf. Proc. Syst.*, 12: 526-532.
- Wikipedia at, Retrieved from: <http://en.wikipedia.org/wiki> (Accessed on: Feb 29, 2012).
- Zhan, Y. and D. Shen, 2005. Design efficient support vector machine for fast classification. *J. Pattern Recogn. Soc. Pergamon*, 38: 157-161.