Trends in Applied Sciences Research 7 (8): 663-672, 2012
ISSN 1819-3579 / DOI: 10.3923/tasr.2012.663.672

Error Detection of Personalized English Isolated-Word Using Support Vector Machine

[1,2]Tiong Sieh Kiong, [1]Abu Bakar Hasan, [1]Johnny Koh Siaw Paw and [3]David Yap Fook Weng

[1]Center of System and Machine Intelligence, Universiti Tenaga Nasional, 43000, Kajang, Selangor, Malaysia
[2]Power Engineering Centre, Universiti Tenaga Nasional, Kajang, 43000, Selangor, Malaysia
[3]Faculty of Engineering, Universiti Teknikal Malaysia Melaka, 76109, Melaka, Malaysia

Corresponding Author: Tiong Sieh Kiong, Center of System and Machine Intelligence, Universiti Tenaga Nasional, 43000, Kajang, Selangor, Malaysia Tel: +603 8921 2282 Fax: +603 8921 2116

ABSTRACT

A better understanding on word classification could lead to a better detection and correction technique. In this study, a new features representation technique is used to represent the machine-printed English word. Subsequently, a well-known classification type of artificial intelligent algorithm namely Support Vector Machine (SVM) is used to evaluate those features under two class types of words with proper segregation of correct and erroneous words in two data sets. Our proposed model shows good performance in error detection and is superior when compared with neural networks, Hamming distance or minimum edit distance technique; with further improvement in sight.

Key words: Error detection, support vector machine, English isolated-word, classification

INTRODUCTION

Feature selection, also known as subset selection or variable selection, is a process commonly used in machine learning, whereby a subset of the features available from the data are selected for application of a learning algorithm (Kukich, 1992; Blum and Langley, 1997). Feature selection can be a powerful tool for simplifying or speeding up computations and when employed appropriately it can lead to a little loss in classification quality (Weston *et al*., 2011). Selecting an optimal set of features is in general difficult, both theoretically and empirically (Dasgupta *et al*., 2007). For practical supervised learning algorithms, the selection is for a satisfactory set of features instead of an optimal set. In machine learning, satisfactory features selection is typically done by cross-validation that is, by looking at which feature sets that could provide a reasonable realization error.

A distinction must be made between the tasks of error detection and correction (Drucker *et al*., 1996). Efficient techniques such as n-gram analysis and lookup tables have been devised for detecting strings that do not appear in a given word list, dictionary or lexicon. Many existing spelling correctors exploit task-specific constraints and focus on isolated words and uses different techniques such as rule-based and minimum edit distance technique (Hsu *et al*., 2003).

We propose a technique using Support Vector Machine (SVM) to detect any error in an isolated English word and propose another possible correct word as the solution to replace the incorrect word. We use Support Vector Classification (SVC) to detect an error in a given word and finally in order to find a possible correct word as the replacement we use Support Vector Regression (SVR). The main objective of this study is to a better technique that could detect and correct an incorrect-isolated English word and that could be used as an alternative technique in the near future.

THEORY ON SUPPORT VECTOR MACHINES

Basically, Support Vector Machines (SVM) is based on the Structural Risk Minimization principle (SRM) from the statistical learning theory (Chang and Lin, 2007). 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 (Nagi *et al.*, 2008). 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 such as getting trapped into the local minimum in neural network (Zhan and Shen, 2005) as shown in Fig. 1.

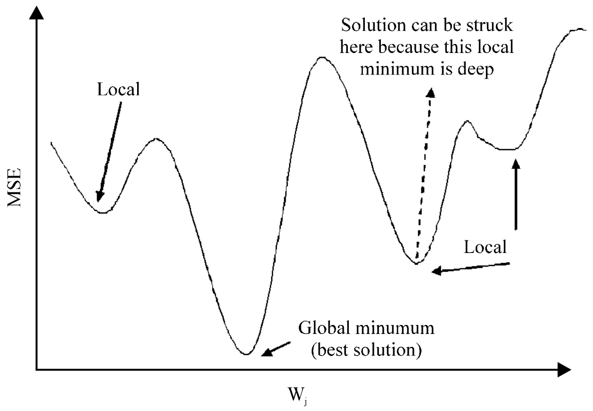
Support vectors in SVM can be categorized into two types: the training samples that exactly locate on the margin of the separation hyperplane and the training samples that locate beyond their corresponding margins, whereby the later is regarded as misclassified samples (Wright, 2011). Given training data consisting of labeled vectors represented by $\{x_i, y_i\}$ for i = 1, 2 ...., m where, $x_i \epsilon R^n$ 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 (Kukich, 1992):

$$w.x_i+b = 1 \quad \text{if} \quad y_i = +1 \tag{1}$$

$$w.x_i+b= -1 \quad \text{if} \quad y_i = -1 \tag{2}$$

A hyperplane which divides the data is to be determined. This amounts in determining w and b so that:

$$y_i(w.x_i+b) = 0, \quad \text{for} \quad i = 1, 2,\ldots\ldots, m \tag{3}$$



Solution can be struck here because this local minimum is deep
Local
MSE
Local
Global minumum (best solution)
W_j

Fig. 1: The phenomenon of “local minimum” in a neural network

If a hyperplane is satisfied, then two classes are known to be linearly separable. Then Eq. 3 is written as:

$$y_i(w.x_i+b) = 1 \tag{4}$$

If the data is not linearly separable, another parameter called the slack variable ξ_i for i = 1, 2, ……, m is introduced with ξ_i>10 such that Eq. 4 can be represented as:

$$y_i\ (w.x_i+b)\ -1+\xi_i = 0 \tag{5}$$

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

$$\min \left[\frac{1}{2} \mathrm{IIwII2} + C\Sigma^{m}_{i=1}\ \xi_i\right] \tag{6}$$

The first term in Eq. 6 above controls the learning capacity while the second term controls the number of misclassified points. The penalty 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 or decision function is:

$$f(x) = \mathrm{SIGN}\ [\Sigma^{m}_{i=1}\ y_i\ \alpha_i(w.x_i+b\ )] = \mathrm{SIGN}\ [\Sigma^{m}_{i=1}\ y_i\ \alpha_i K(x_i,\ x_j)+b] \tag{7}$$

whereby, α_i is the support vector of nonnegative Langrange multipliers.

Parameters α_i and b are to be determined by SVM's learning algorithm (Vapnik, 1998) during the classification process. Parameter y_i and $K(x_i, x_j)$ is 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, in order to solve nonlinear problems SVMs use a kernel function to allow a better fitting of the hyperplane to more general datasets. There are a few types of kernel functions proposed by Vapnik (1998), for example, the Radial Basis Function (RBF) kernel which nonlinearly maps samples into a higher dimensional space. The RBF kernel has less numerical difficulties and has the property defined by $K(x_i, x_j) = \exp(-\gamma \mathrm{I}x_i\text{-}x_j\mathrm{I}^2)$. The kernel bandwidth parameter γ controls the scaling of the mapping (Vivekan, 2010) and C for the RBF kernel are pre-determined by v-fold cross validation or grid-search (Cristiani and Taylor, 2000). If the number of features is large, linear kernel was selected instead of RBF kernel because nonlinear mapping does not improve the SVM performance. A recent result shows that if RBF is used with model selection, then there is no need to consider the linear kernel (Gunn, 1997). At the moment SVM usage generally covers data classification and regression only (Drucker *et al*., 1996; Kohavi and John, 1997).

METHODOLOGY ON SVM CLASSIFICATION 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

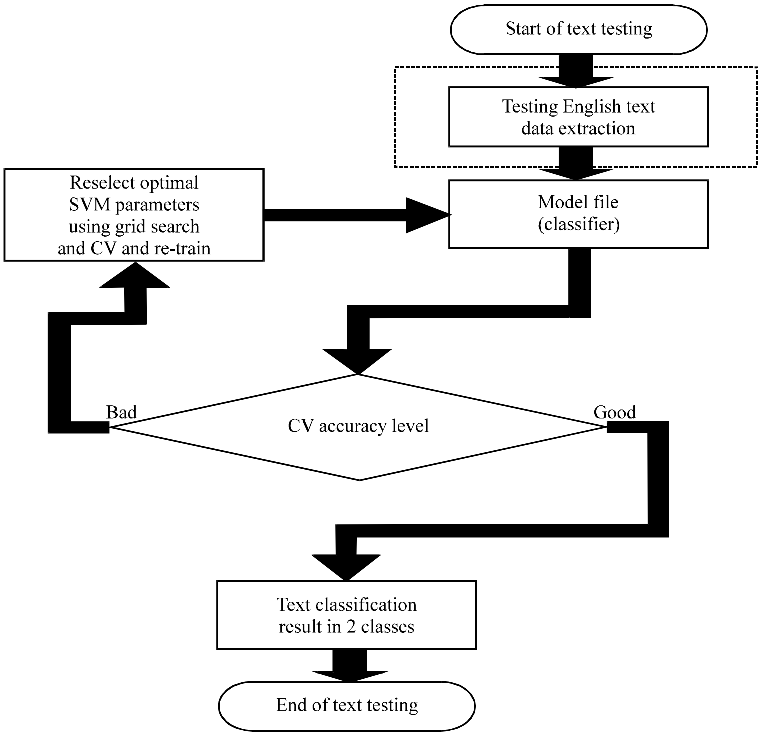

Start of text testing
Testing English text data extraction
Reselect optimal SVM parameters using grid search and CV and re-train
Model file (classifier)
Bad
CV accuracy level
Good
Text classification result in 2 classes
End of text testing

Fig. 2: Flowchart framework of word classification

study, we are proposing a model to represent a word to meet the SVM classification requirements (Fig. 2), with a maximum of ten 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 (w1) and last letter (wn), position-vowel factor (pvf), sum of position-consonant-vowel factor (pcf), position factor (pf) 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: len = nv+v

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: $w = w_1+w_2+\ldots\ldots.+w_n$

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 position factor is the sum of position-vowel factor and the sum of position-consonant-vowel factor: pf = pcf+pvf

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, for example consider the word reddish, Table 1 illustrates the attributes value with the help of Rule 1, 2, 3 and 4.

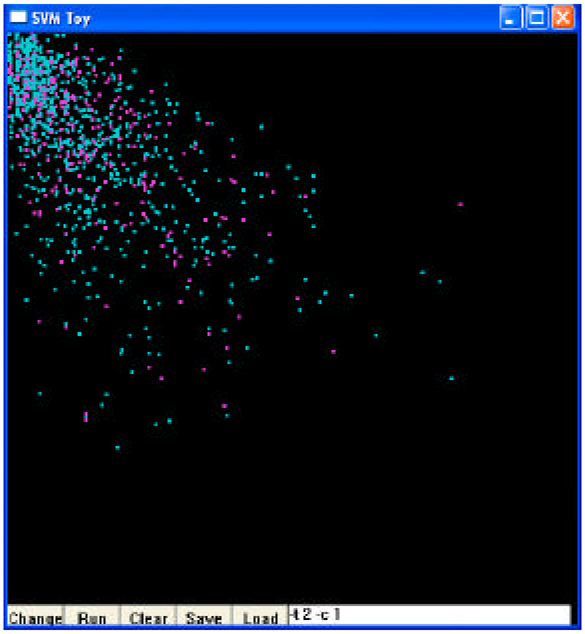
We used LibSVM (Hsu *et al.*, 2003; Chang and Lin, 2007) which is a SVM learning tool package, for the training and testing process of the normalized words. The five rules above will

Table 1: Attribute values for 'reddish'

Attribute	Value	Attribute	Value
nv	5	wn	8
v	2	pvf	56
len	7	pcf	216
w	67	pf	271
w1	18	typ	0

```
1. Set all the training and testing data (i.e., the words) attributes to ten
2. Normalize the raw data
3. Find the best value for parameters C and γ during cross
4. Get the support vectors from the classification process
5. Prompt the user to enter the testing data
6. Calculate the decision function
7. Prints the class for each of test data
```

Fig. 3: Pseudo-code of training and testing engine



t 2 -c 1

Fig. 4: Training data on SVM-toy

mathematically play an important function in the classification of each word. For the classification, we are proposing a 2-steps simplified data processing procedure as follows (Fig. 3):

Step 1: Change the raw data into SVM format. Select the cross-validation parameter (v) equal 10 and the kernel function (RBF). By choosing v = 10, LibSVM package (Chang and Lin, 2007) will automatically divides the training data into ten different parts with basically equal numbers of data. Train the ten parts of these training data (Fig. 4) to determine the best value for parameters C and γ. The best values refer to the highest efficiency

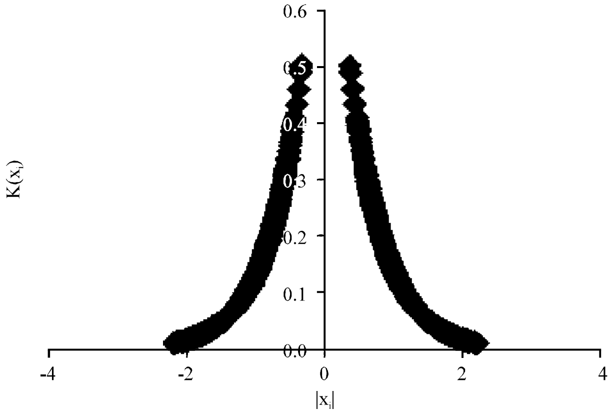


Fig. 5: Kernel $K(x_i)$ vs. $|x_i|$ with $\sigma = 0.5$

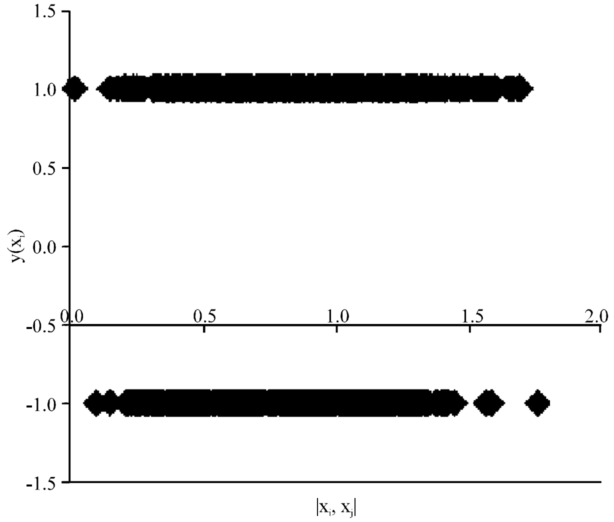


Fig. 6: Decision function $y(x_i)$ vs. $|x_i, x_j|$

percentage in classifying the training data. After the best (C, γ) was chosen, the whole training set is trained again to generate the final classifier or model (Fig. 5), in which support vectors from the trained data are being determined automatically by LibSVM package (Fig. 6). These support vectors play an important function in classifying those trained data

Step 2: Test the unknown data to obtain the final classification efficiency which represents how good the model obtained in step 1 in identifying other data

RESULTS AND DISCUSSION

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, 2 and 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 v = 10. This time we obtained the testing accuracy of 77% with the values of best (C, γ) = (1, 1) and number of attributes equals seven. In the second work, we increased the number of attributes and training words but maintaining both parameters C, γ and v; giving a much better test accuracy value of 79%, as shown in Table 2. Table 3 shows that as we increased the number of attributes used, subsequently we observed the training and test accuracies remained constant at C = 1, γ = 1 and v = 10. Meaning that for our case study, the accuracy did not depend on the number of attributes of a text. This finding is in line with Kukich (1992) work, whereby he mentioned that there was 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) range as proposed by Hsu *et al.* (2003). 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. Wright (2011) has proposed the 4S approach, whereby for a good spelling skill, one must understand the importance of vowels in any syllable in a word. 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 their selected attributes was mainly based on the character size or the

Table 2: Effects of number of attributes, training and test word on test accuracy

No. of attributes used	No. of training word	No. of test word	Test accuracy (%)
7	1117	478	77
10	1142	451	79

Table 3: Relationship between number of attributes used with accuracy

No. of attributes used	No. of training word	No. of test word	Training accuracy (%)	Test accuracy (%)
3	1142	451	69.58	78.44
6	1142	451	69.58	78.44
8	1142	451	70.00	79.00
10	1142	451	70.00	79.00

Table 4: Comparison with Kukich (1992) study

Technique	1142-word lexicon (%)
Test accuracy	
Minimum edit distance (use grope)	62
Similarity key (bocast token reconstruction)	78
Simple N-gram vector distance (use hamming distance)	68
Probabilistic (use Kernighan-Church-Gale error probe)	78
Neural network (use back propagation classifier)	75
Support vector machines (use radial basis function)-proposal technique	79

Table 5: Comparison with UCI benchmark repository

Techniques	Cross-validation	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 6: Partial calculation for Kernel function $K(x_i, x_j)$ for i = 1 and 2, $\gamma = 1$

Iteration i	α_i y_i	Kernel $K(x_i, x_j) = \exp(-\gamma I x_i - x_j I^2)$
1	0.9886678441001486	$EXP[-SQRT\{(0.033282905-0.148260212)^2+(0.081141998-0.211119459)^2+(0.064791562-0.189854345)^2+(0.33333333-0.444444444)^2+(0.25-0.5)^2+(0.25-0.4375)^2+(0.18636364-0.363636364)^2+(0.2-0.48)^2+(0.45833333-0.541666667)^2+(0-0)^2\}]$
2	0.6392899539619362	$EXP[-SQRT\{(0.019667171-0.148260212)^2+(0.072877536-0.211119459)^2+(0.054746359-0.189854345)^2+(0.22222222-0.444444444)^2+(0.25-0.5)^2+(0.1875-0.4375)^2+(0.21818182-0.363636364)^2+(0.96-0.48)^2+(0.70833333-0.541666667)^2+(0-0)^2\}]$

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 seven to ten different behavioral features in representing an English word and we used 10-fold cross-validation. Referring to Table 4, we observed that our work managed to produce the highest accuracy; meaning that SVM could provide us the best tool when comes to classification of isolated-word error in texts. Table 5 shows the comparison with UCI benchmark repository on the related work.

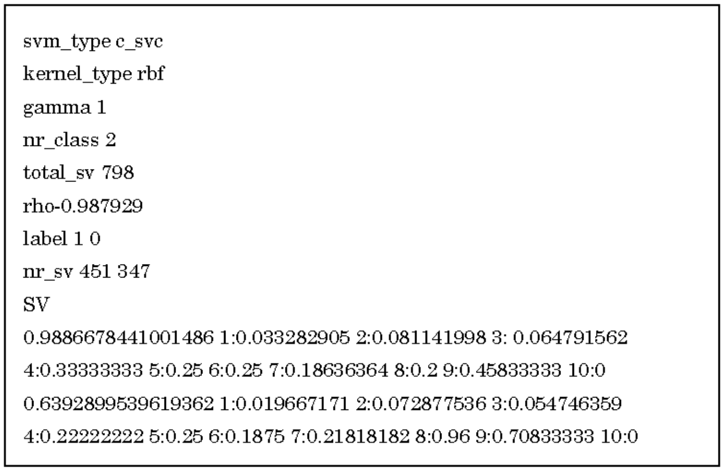
During the classification process, one of the file produced was a model file (Fig. 7) from which we managed to obtain the parameter b = -rho = 0.987929, γ = gamma = 1 and the summation of Eq. 7 can be partly shown as in Table 6 for a set of prediction data $X_{j1} = 0.148260212$, $X_{j2} = 0\ .211119459$, $X_{j3} = 0.189854345$, $X_{j4} = 0.444444444$, $X_{j5} = 0.5$, $X_{j6}=0.4375$, $X_{j7} = 0.363636364$, $X_{j8} = 0.48$, $X_{j9} = 0.541666667$, $X_{j10} = 0$. Parameter m in Eq. 7 equals to the number of support vectors that is, m = i = total_SV = 798. Obviously, The RBF kernel is 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 the decision function $f(x_i)$ for i = 1 becomes:

$$f(x_1) = [0.9886678441001486\ (EXP[-SQRT\{(0.033282905-0.148260212)^2+ (0.081141998-0.211119459)^2+(0.064791562-0.189854345)^2+(0.33333333-0.444444444)^2+ (0.25-0.5)^2+(0.25-)\ 0.4375)^2+(0.18636364-0.363636364)^2+(0.2-0.48)^2+(0.45833333-0.541666667)^2+(0-0)^2\}])+(-0.987929)] = 0.586585013 \tag{8}$$

Table 7: LibSVM classification selected results

Original label	Test data or word	f (x_i)	Class obtained	Comment
1	hazardous	SIGN (1.09336784)	+1	Classified correctly
0	hazardous	SIGN (-0.882304417)	-1	Classified correctly
0	hazardous	SIGN (-0.905236828)	-1	Classified correctly



```
svm_type c_svc
kernel_type rbf
gamma 1
nr_class 2
total_sv 798
rho-0.987929
label 1 0
nr_sv 451 347
SV
0.9886678441001486 1:0.033282905 2:0.081141998 3: 0.064791562
4:0.33333333 5:0.25 6:0.25 7:0.18636364 8:0.2 9:0.45833333 10:0
0.6392899539619362 1:0.019667171 2:0.072877536 3:0.054746359
4:0.22222222 5:0.25 6:0.1875 7:0.21818182 8:0.96 9:0.70833333 10:0
```

Fig. 7: Sample of model output file

Similarly, the other values for the decision function for i = 2-798 can be mathematically calculated as $f(x_2) = 0.224201394$ and $f(x_{798}) = -0.309491908$. Hence:

$$f(x) = \text{SIGN}\ [\Sigma_{i=1}^{798} y_i \alpha_i K(x_i, x_j)+b] = \text{sign}\ [f(x_1)+f(x_2)+\dots\dots\dots+f(x_{798})+b] = \text{SIGN}\ [0.586585013+0.313526685+\dots\dots\dots+ (-0.523892553)+(0.987929)] = \text{SIGN}\ (0.913826625) = +1 \quad (9)$$

Graphically f(x) is a mapped point on the positive side of the hyperplane of the feature space, meaning that the 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. Figure 6 represents the overall plot for 478 test data which have been successfully classified into two different classes. Table 7 shows the classification results on three other test data or words.

CONCLUSION

In this study, we have demonstrated in detail the ability of SVM in classifying isolated-word error for English word and one should consider SVM when dealing with classification of data. SVM technique produces the better accuracy for English isolated-word error detection when compared to the other techniques mentioned in this study.

REFERENCES

Abu Zarim, Z.A., A.B. Abd Ghani, K.S. Yap and S.K. Tiong, 2008. Underground power cable failure determination by using support vector machine (SVM). Proceedings of the 17th Conference of the Electric Power Supply Industry, CEPSI2008, 27-31 October 2008, Macau, China.

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

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

Chang, C.C. and C.J. Lin, 2007. LIBSVM-A library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html

Cristiani, N. and J.S. Taylor, 2000. An Introduction to SVM and Other Kernel-based Learning Methods. Cambridge University Press, UK., Pages: 189.

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V. and M.W. Mahoney, 2007. Feature selection methods for text classification. Proceedings of the 13th ACM Int. Conference on Knowledge Discovery and Data Mining, 12-15 August 2007, California, USA, pp: 230-239.

Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A. and V. Vapnik, 1996. Support vector regression machines. Adv. Neur. Inf. Process. Sys., 9: 155-161.

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, 2003. A practical guide to support vector classification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

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

Kohavi, R. and G.H. John, 1997. Wrappers for feature subset selection. Artif. Intell., 97: 273-324.

Kukich, K., 1992. Techniques for automatically correcting words in text. ACM Comput. Surveys, 24: 377-439.

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

Nagi, J., Mohammad, A.M., Yap, K.S., Tiong, S.K. and S.K. Ahmed, 2008. Non-technical loss analysis for detection of electricity theft using support vector machines. Proceedings of the Second IEEE Int. Conference on Power and Energy (PECON). 1-3 Dec 2008, Johor Bahru, pp: 907-912.

Rani, R., Dhir, R. and G.S. Lehal, 2011. Identification of printed Punjabi words and English numerals using Gabor features. World Acad. Sci. Engin. Technol., 73: 392-395.

Vapnik, V.N., 1998. Statistical Learning Theory. 1st Edn., John Wiley and Sons, New York.

Vivekan, M., 2010. Internal technical reports. UNITEN, Malaysi.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Puggio, T. and V. Vapnik, 2011. Feature selection for SVM. http://www.cs.ucl.ac.uk/staff/M.Pontil/reading/featsel.pdf

Wright, K.W., 2011. Vowels are vital, star educate. The Star Newspaper.

Zhan, Y. and D. Shen, 2005. Design efficient support vector machine for fast classification. Pattern Recogn., 38: 157-161.