PERFORMANCE COMPARISON OF ONLINE HANDWRITING RECOGNITION SYSTEM FOR ASSAMESE LANGUAGE BASED ON HMM AND SVM MODELLING

Deepjoy Das, Rituparna Devi, SRM Prasanna, Subhankar Ghosh, Krishna Naik

Department of EEE, IIT Guwahati, Assam
{deepjoy2002@gmail.com, rituparna.sarma5@gmail.com, prasanna@iitg.ernet.in, ghoshsubhankar@gmail.com, krishnanaik.35@gmail.com}

ABSTRACT

This work emphasises on the development of Assamese online character recognition system using HMM and SVM and performs a recognition performance analysis for both models. Recognition models using HTK (HMM Toolkit) and LIBSVM (SVM Toolkit) are generated by training 181 different Assamese Stokes. Stroke and Akshara level testing are performed separately. In stroke level testing, the confusion patterns of the test strokes from HMM and SVM classifiers are compared. In Akshara level testing, a GUI (provided by CDAC-Pune) which is integrated with the binaries of HTK/LIBSVM and language rules (stores the set of valid strokes which makes a character) are used, manual testing is done with native writers to test the Akshara level performance for both models. Experimental results show that the SVM classifier outperforms the HMM classifier.

KEYWORDS

Support Vector Machines, Hidden Markov Models, Handwriting Recognition, Assamese, LIBSVM, HTK

1. INTRODUCTION

Of the various handwriting recognition systems available, there exists two basic handwriting recognition domains distinguished primarily by nature of the input signal-online and offline. In offline system the digitised information is in the static form whereas in the online system, information is acquired during production of the handwriting using equipments such as Tablet PC which captures the trajectory of the writing tool. The information captured undergoes some filtration, pre-processing and normalisation process after which the handwriting is segmented into basic units which are usually a character or part of a character. Finally each segment is classified and labelled. In our system, we examine the effectiveness of using Hidden Markov Models (HMM) and Support Vector Machines (SVM) for modelling the classifier. HMM has been used for Bangla [2], Telugu [3], Tamil [4], Malayalam [5] and in previous works for Assamese [6] [7]. Support vector machines (SVMs) have also been used in [8] for Telegu and Devnagiri scripts while [9] compares the performance between systems developed using HMM and SVM for Telegu script. The classifiers are built individually using HMM and SVM and the recognition accuracies of both the systems are analysed for comparison. In our work, the two coordinate trace

Natarajan Meghanathan et al. (Eds) : ICCSEA, SPPR, VLSI, WiMoA, SCAI, CNSA, WeST - 2014
pp. 87–95, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.4717

between one pen down to pen up is taken as the basic unit and is termed as stroke and a set of 181 characters have been classified as valid stroke after detailed analysis of the handwriting of the Assamese language.

1.1 Assamese language and Assamese character database

The Assamese language is an Indo Aryan language and is used in the North Eastern part of India. Sanskrit is the mother language of Assamese and though it has its own script, its phonetic character set and behaviour is derived from Sanskrit. It currently has a total of 11 vowels, about 41 consonants, 10 numerals and a number of conjuncts, vowel modifiers, consonant modifiers and other symbols. The vowels, consonants and numerals are previously defined in the language. However to obtain a set of valid conjuncts, the Assamese script has been thoroughly scrutinised as the frequency of using conjuncts is not well defined in the script. Eventually the Assamese script of 241 aksharas is formed consisting of 11 vowels, 41 consonants, 147 conjuncts, 10 numerals, 10 vowel modifiers, 2 consonant modifiers and 20 additional symbols after analysing the commonly used conjuncts obtained by scanning the pages of Hemkosh Dictionary written by Hem Chandra Barua and comparing it with a list of 147 conjuncts prepared by the Resource Centre for Indian Language Technology Solutions (RCILTS), Indian Institute of Technology, Guwahati. These are then sent for data collection. The list of 241 symbols is depicted in figure 1.

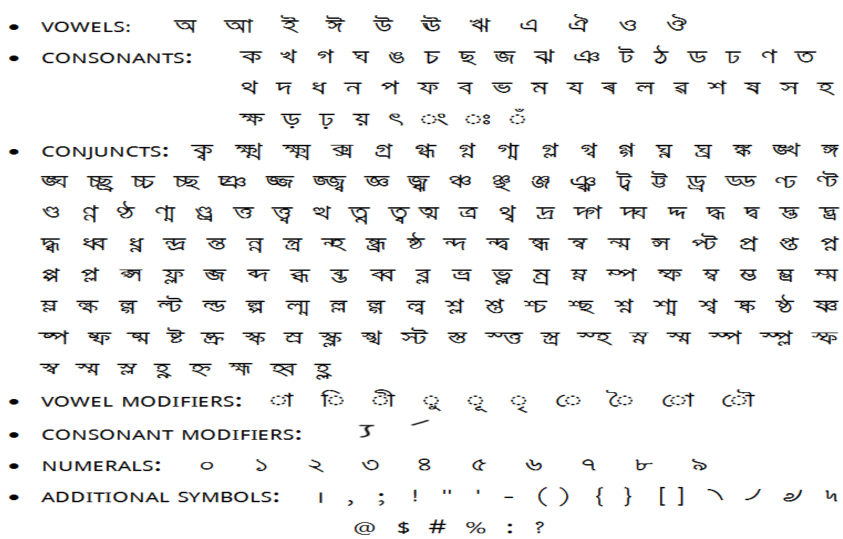

Figure 1: Assamese Aksharas

The data is collected with a HP Tablet PC with a sampling rate of 120 Hz and the captured information contains values of horizontal & vertical coordinates along with writer information. Finally considering writer feedback as well, a final list of 147 Assamese symbols is created consisting of 11 vowels, 41 consonants, 55 conjuncts, 10 numerals, 23 special symbols and 5 characters whose shape change on addition of modifiers.

1.2 Characters of Assamese handwriting

For classification and labelling, the handwriting data is segmented into basic components called strokes which are generally a whole character or part of the character. In our work, the coordinate trace between one pen down to pen up is taken as the basic component or stroke. However as Assamese handwriting is a case of cursive handwriting, several components may be merged or even broken down by infrequent writers. Thus data grouping is done in three ways namely strokes, substrokes and suprastrokes. The strokes consist of a typical basic component of aksharas agreed naturally by majority of non-cursive writers. The substrokes contained components are formed by merging several components or strokes. Again if the infrequent writers break the components of the stroke into more than one component then splitted components form the suprastrokes. A list is prepared combining all the above strokes of different data groups and we have a final list of 203 distinct strokes in Assamese handwriting. The final strokes list is depicted below in Figure 2.

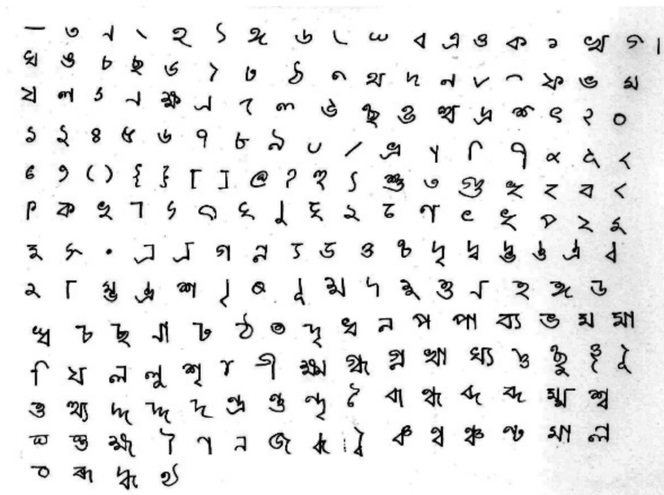

Figure 2 Isolated Assamese strokes

2. CHARACTER RECOGNITION SYSTEM

The schematic block diagram of Assamese Online Stroke Recognition system using HMM & SVM Modelling is shown below in Figure 4.

2.1 Database

The Assamese data set consists of a set of 203 isolated strokes or basic components as shown in Figure 2. A set of 147 Assamese Aksharas have been finalised to acquire the stroke data required to design a robust stroke recogniser. The Akshara examples have been collected from 100 users in two sessions in an HP Tablet PC using an open source tool developed by HP with a sampling rate of 120 Hz. From each of the Akshara sample, the basic component or stroke is extracted from one pen down to pen up which results in about 1000 examples for each stroke of which 50 % are used for training and 50 % for testing. During data collection no limitation or restriction is enforced on the style of writing and hence, we have a large variation among the samples of a given stroke.

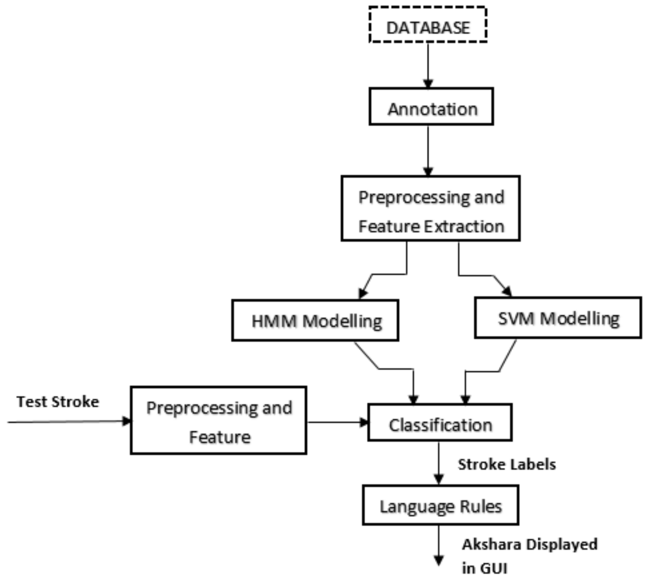

Figure 3: Assamese Online Stroke Recognition System Using HMM & SVM Modelling

2.2 Annotation

Annotation refers to the labelling of collected database in order to arrange them into groups of analogous patterns which in the milieu of our work are the strokes, substrokes and suprastrokes of Assamese language. The classifiers are then trained with these patterns which are then used for recognition purpose. The desired outcome after annotation is a database completely labelled at the sentence, word, character and stroke labels. The annotated data is analysed to finalise the set of strokes, substrokes and suprastrokes and finally only those patterns or strokes are retained which are used by more than 5 % users.

2.3 Pre-processing and Feature Extraction

The pre-processing stage consists of size normalization, smoothing, interpolation of missing points, removal of duplicate points and resampling of the captured coordinates [10].

2.3.1 Size Normalisation

The size of each individual data sample is normalised by scaling the pattern both horizontally and vertically [11]

2.3.2 Smoothing

Smoothing excludes the noise captured during the data collection process and performed using a moving average filter of size three. Each pattern is smoothed both in horizontal and vertical directions discretely [12]

2.3.3 Removal of duplicate points

Duplicate points do not contain any information and only cause data redundancy and hence these points are removed before feature extraction [12]

2.3.4 Resampling

Resampling eliminates the disparities in the data due to the writing speed of the writers. It is performed by linear interpolation of missing points which results in a sequence of equidistant points [12]

2.4 Feature Extraction

The pre-processed horizontal & vertical coordinates and their first and second derivatives can be used as features for the modelling of the stroke classifier. The first derivative gives the change and the second derivative gives the change of change in horizontal and vertical coordinates. The first derivative is calculated to observe the change in the trajectory at current point. The second derivative is calculated in order to examine the change of change in trajectory at current point. A window size of two is considered in both the cases. The method of extracting feature vectors is identical for both the classification models used in our work.

2.5 Classification Models

The efficiency of HMM and SVM are studied for developing the stroke classifiers.

2.5.1 HMM Modelling and testing

HMM models a doubly stochastic process, one observable and the other hidden [14]. In our work, the sequence of feature vectors from the online handwriting is the visible stochastic process and the underlying hand movement is the later. In the present work,for modelling each stroke, one left to right, continuous density HMM is developed. The left to right structure is used supposing distinctive directions of handwriting movements [15]. After collected database is annotated at stroke level, six dimensional features are extracted from the pre-processed coordinates. 203 strokes are finalised during script analysis of the Assamese language and hence 203 HMM models are built for each of the 203 strokes. All the test examples corresponding to each stroke class are tested against all the stroke models and if misclassification arises due to resemblance in pattern shape between two strokes those stroke classes are merged. Hence the final stroke classifier is developed with 181 stroke classes. The HMM models are trained using 7 states and 20 mixtures and HMM Toolkit (HTK) is used for training and testing.

A. HMM Training

The feature vectors described in the previous section are used for training the HMM which comprises of a set of states and the alterations linked with it and are trained using Baum-Welch re-estimation or expectation maximization (EM) approach [14]. In this procedure an initial model is taken and improved model parameters are re-estimated using the given set of feature vectors. The most recent model is the initial model for the next iteration and again re-estimation is done using the same set of feature vectors. This procedure is recurring until model parameters become static and the model of the last reiteration is stored as the model for the given class [12]. The process is repeated for all stroke classes.

B. HMM Testing

During testing the class information for the examples that are unknown to the trained model are found out. The likelihood probability of the given test example against each of the trained HMM models are determined and the model with the highest likelihood is theorised as the class. The process is repeated for all the testing examples and the class information is noted [12], [15].

2.5.2 SVM Modelling

Similar to HMMs, if given a set of training samples, SVMs will attempt to build a model. Each training data instance is marked as belonging to one of two categories. The SVM will attempt to separate the data instances into those two categories with a p-1 dimensional hyperplane, where p is the size of each data instance. This model can then be used on a new data instance to predict which category it would fall onto. The maximum margin hyperplane can be represented as [17]

$$y(x) = b + \sum \alpha_i y_i K(x(i), x)$$

Vector x is a test case and y_i is the class value of the training example $x(i)$. In the equation, the parameters of the hyper plane are b and α_i. b is a real constant, and α_i are non-negative real constants. The function $K(x(i), i)$ is a kernel function and SVMs are powerful in the sense that one can substitute different kernel functions. The four basic kernel functions are [18]

1. Linear: $K(x_i, x_j) = x_i^T x_j$
2. Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
3. Radial(RBF): $K(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}, \gamma > 0$
4. Sigmoid: $K(x_i, x_j) = (\gamma x_i^T x_j + r)$

The classifier can be constructed as follows: [19]

$$w^T \varphi(x(i)) + b \geq 1, \text{if } y_i = 1$$
$$w^T \varphi(x(i)) + b \leq 1, \text{ if } y_i = -1$$
$$y_i \left[w^T \varphi(x(i)) + b \right] \geq 1, \ i = 1, \text{K K}, N$$

Where $\varphi(-)$ is a nonlinear function that maps the given inputs into some higher dimensional space. In case we cannot find the separating hyperplane in this space, we introduce additional variables: ξ_i, Where $i = 1, \ldots.., N$. After this, we will attempt to solve this minimization problem:

$$\min_{w,b,\xi_i} J(w, \xi_i) = \frac{1}{2} w^T w + c \sum_{i=1}^{N} \xi_i \ \text{ s.t. } \ y_i \left[w^T \varphi(x(i)) + b \right] \geq 1 - \xi_i \ \ \xi \geq 0, \ \ i = 1, \text{K}, N$$

The solution to the above model will be the optimal separating hyperplane.

A. SVM Training and Testing

We use the LIBSVM defaults "radial basis" kernel for mapping a given set of input vectors into a higher dimensional space. The pre-processing step involves extraction of four dimensional feature vectors namely the horizontal & vertical coordinates and their first derivatives. The second derivatives are not used as they have been found to give reduced recognition accuracy. As SVM works with fixed sized vectors, we choose 60 equidistance handwritten points, which span the whole handwritten curvature (choose more points in high curvatures).

3. GRAPHICAL USER INTERFACE (GUI)

The GUI of the testing tool has been developed by Centre for Development of Advanced Computing Graphics and Intelligence based Script Technology Group (CDAC), Pune, India and is provided with API Calls. A Particular API call was used to get certain service out of the GUI. We have integrated the GUI with our HMM and SVM training models separately one at a time, along with valid set of language rules (stored in text file) using a DLL.

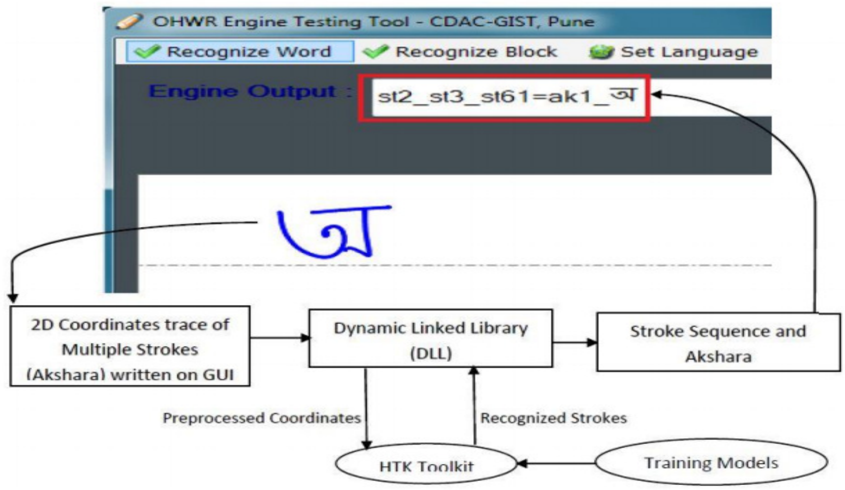

Figure 4: Block Diagram of Akshara Recognition with GUI and DLL. Akshara 1 recognized with stroke 2, 3 and 61.

When a stroke is written on the GUI, the parameters of API provide the basic data like 2-dimensional coordinate traces of the stroke. The dynamic linked library when provided with raw handwritten trace, it initiates pre-processing for refinement of the 2-dimensional trace and then performs classification tasks with the integrated HMM testing model and outputs the set of labels of recognized stroke. The set of strokes are then checked with language rules to verify whether there exist a valid Akshara for the respective strokes. If yes, then output the valid Akshara in GUI text box.

4. COMPARISON RESULTS BETWEEN HMM AND SVM CHARACTER RECOGNITION SYSTEM

During testing, log likelihood values are obtained from HMM classifier while SVM classifier gives probability estimates as output. The output from both the stroke recognizers are then compared using two approaches. In the first approach, the confusion matrix is obtained from both the HMM classifier and SVM classifier and the confusion patterns are analysed. The confusion matrix for the first 10 classes out of the 181 classes obtained using HMM is shown in Figure 6 and the confusion matrix for SVM classifier is shown in Figure 7. In the second approach, users are allowed to write the stroke patterns in the testing tool obtained by integrating the GUI provided by CDAC, Pune once with the HMM classifier and once with the SVM classifier and accuracy of both the classifiers are studied manually. The developed stroke classifier gives average recognition accuracy of about 94 % in case of HMM and 96 % in case of SVM. The akshara level average performance is 84.67 % in case of HMM and 86.23 % in case of SVM

stroke no.	st1	st2	st3	st4	st5	st6	st7	st8	st9	st10
st1	**69.68**	0	0	8.30	0	1.21	0	0	0.75	0.15
st2	0	**92.58**	0	0	0	0	0	0	0	0
st3	0	0	**94.67**	0	0	0	0	0	0	0
st4	0.15	0	0	**94.93**	0	0.31	0	0	0.77	0
st5	0	0	0	0	**85.71**	0	0	0	0	0
st6	0.24	0	0	0	0	**86.03**	0	0	0.12	0
st7	0	0	0	0	0	0	**95.06**	0	0	0
st8	0	0	0	0	0	0	0	**90.03**	0	0
st9	0	0	0	0	0	0	0	0	**98.31**	0
st10	0	0	0	0	0	0	0	0	0	**96.98**

Figure 5: confusion percentage matrix for the first 10 strokes using HMM classifier

stroke no.	st1	st2	st3	st4	st5	st6	st7	st8	st9	st10
st1	**69.71**	0	0	4.50	0	0	0	0	0.75	0
st2	0	**92.58**	0	0	0	0	0	0	0	0
st3	0	0	**94.68**	0	0	0	0	0	0	0
st4	0.09	0	0	**95.01**	0	0.31	0	0	0.77	0
st5	0	0	0	0	**85.71**	0	0	0	0	0
st6	0.11	0	0	0	0	**86.58**	0	0	0.12	0
st7	0	0	0	0	0	0	**95.08**	0	0	0
st8	0	0	0	0	0	0	0	**90.03**	0	0
st9	0	0	0	0	0	0	0	0	**98.31**	0
st10	0	0	0	0	0	0	0	0	0	**97.02**

Figure 6: confusion percentage matrix for the first 10 strokes using SVM classifier

5. CONCLUSION

We have observed that the feature vector namely the coordinate trace, first derivate and second derivative works perfectly with HMM based system, however the recognition accuracy reduces significantly when the second derivate is used as a feature in SVM based system. Therefore in SVM based system only the coordinate trace and first derivate is used as a feature. In HMM based system the recognition accuracy reduces if second derivate is not used in the feature set. The SVM based system outperforms HMM based system by 2% in stroke accuracy and 1.56% in akshara case. The performance is almost similar. The performance might improve if we consider a larger set of database than currently used in SVM case.

REFERENCES

[1] N. Arica and F. T. Yarman-Vural, "An Overview of Character Recognition Focused on off-line Handwriting," IEEE Trans. Systems, Man, Cybernetics Part C: Applications and Reviews, vol. 31, no. 2, pp. 216-233, May 2001

[2] S. K. Parui, K. Guin, U. Bhattacharya, and B. B. Chaudhuri, "Online Bangla Handwritten Character Recognition using HMM," in Proc. 19th Int. Conf. on Pattern Recognition (ICIP),, pp. 1-4, Tampa FL, 2008

[3] V. J. Babu, L. Prasanth, R. R. Prasanth, R. R. Sharma, G. V. P. Rao and A. Bharath, "HMM-based online handwriting recognition system for telugu symbols," in Proc. 9th Int. Conf. on Document Analysis and Recogntion (ICDAR), Curitaba, Brazil, 2007, pp. 63-67

[4] K. Shashikiran, K. S. Prasad, R. Kunwar, A. G. Ramakrishnan, "Comparision of HMM and SDTW for Tamil handwritten character recognition." in Proc. Int. Conf. on Signal Processing and Communications,, pp. 1-4, IISc Bangalore, India, 2010

[5] A. Arora and A. M. Namboodiri, "A Hybrid Model for Recognition of Online Handwriting in Indian Scripts," in Proc. of Int. Conf. on Frontiers in handwriting Recognition, pp. 433-438, Kolkata, 2010

[6] G. S. Reddy, P. Sharma, S. R. M. Prasanna, C. Mahanta and L. N. Sharma, “Combined Online and Offline Assamese Handwritten Numeral Recognizer”, in Proc. of 18th National Conference on Communications, pp. 1–5, 2011

[7] G. S. Reddy, B. Sarma, R. K. Naik, S. R. M. Prasanna and C. Mahanta, “Assamese Online Handwritten Digit Recognition System using Hidden Markov Models”, accepted at the Workshop on Document Analyisis & Recognition, 2012

[9] A. Jayaraman, C. Chandra Sekhar and V. S. Chakravarthy, “Modular Approach to Recognition of Strokes in Telegu Script”, in Proc. Of 9th International Conference on Document Analysis and Recognition, pp. 501-505, 2007

[10] V. J. Babu, L. Prasanth, R. R. Prasanth, R. R. Sharma, G. V. P. Rao and A. Bharath, “HMM-based online hanwriting recognition system for telugu symbols,” in Proc. Of 9th Int. Conf. on Document Analysis and Recogntion Brazil, 2007, pp. 63-67.

[11] X.‘Li and D. Y. Yeung, “Online handwritten alphanumeric character recognition using dominant points in strokes”, Pattern Recognition, vol. 30, no. 1, pp. 31-44, 1997.

[12] S.R.M Prasanna, Rituparna Devi, Deepjoy Das, Subhankar Ghosh, Krishna Naik, “Online Stroke and Akshara Recognition GUI in Assamese Language Using Hidden Markov Model” International Journal of Scientific and Research Publication, ISSN 2250-3153

[13] http://htk.eng.cam.ac.uk/

[14] L. R. Rabiner, “A tutorial on Hidden Markov Models and selected applications in speech recognition,” Proc. Of IEEE, vol. 79, no. 2, pp. 257-286, 1989.

[15] G. Siva Reddy, Bandita Sharma, R.Krishna Naik, S.R.M Prasanna, Chitralekha Mahanta, “ Assamese Online Handwritten digit recognition system using Hidden Markov Models” in Proc Of the workshop on Document Analysis and Recognition, pp. 108-113, DAR’12.

[16] Nello Cristianini and John Shawe-Taylor, ―An Introduction To Support Vector Machines And Other Kernel-Based Learning Methods‖, Cambridge University Press, 2000.

[17] K. Kim, Financial Time Series Forecasting Using Support Vector Machines, Elsevier, March 2003

[18] C. Hsu, C. Chang, C. Lin, A Practical Guide to Support Vector Classification, National Taiwan University, 2003

[19] Z. Hua, Y. Wang, X. Xu, B. Zhang, L. Liang, Predicting Corporate Financial Distress Based on Integration of Support Vector Machine and Logistic Regression, Expert Systems with Applications, 2007