Statistical Feature Extraction to Discriminate Various Languages: Plain and Crypt

Neelam Verma, S. S. Khan, Shri Kant
Scientific Analysis Group,
Defence R & D Organization,
Delhi
E-mail: neelam123_v@yahoo.co.in, shehrozkhan@rediffmail.com

ABSTRACT
The present paper deals with the problem of identification among English, romanized Hindi & Punjabi language directly from its plain bit stream. A novice feature extraction technique namely BTR has been discussed & the description of each feature component has been highlighted. An attempt has also been made to identify above-mentioned languages from their cipher bit stream obtained through a block cipher DES & stream cipher Geffe Generator.

Keywords: crypto algorithm, cryptanalysis, linguistic, BTR, feature extraction, functional approximation, neural networks

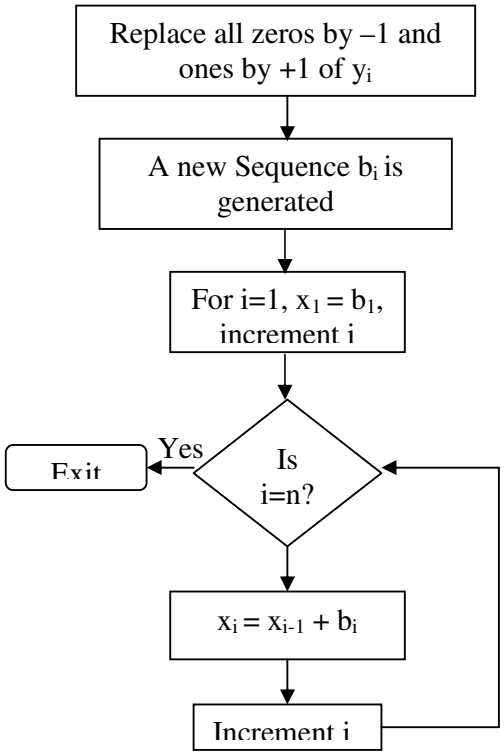
1. INTRODUCTION
Security of information exchanged through communication network and online computers is a formidable task. The plain messages of any language are changed to binary bit stream through some coding algorithm and then encrypted through some crypto algorithm to communicate upon the open channel. When a cryptanalyst receives a junk of bits from this channel, he would like to find which language is being used in this bit stream. The problem gets worse and the success rate degenerates when the communication channel is flooded with encrypted form of these language messages. Since every language has got its signature, so he might like to figure out, which language is surging on the channel? If this encrypted set of languages is identified, somehow, he can consult linguistic experts, use heuristic rules and may cryptanlayze the message efficiently and may be successful in avoiding the brute force attack. But the main problem still remains, the identification of languages for the encrypted bit stream. Knowing the complexity of the problem, advanced techniques using mathematics, computer science and information technology are needed to create new theory and models, to build the architecture and platforms for successful completion of the language identification task. In the recent times language identification is gearing up quite fast and researchers are showing renewed and increased interest. A lot of research is carried out in this area using different techniques.

In the present paper the problem of identification among plain bit stream of three languages viz English and romanized Hindi & English and romanized Punjabi have been successfully carried out with the help of Pattern Recognition (PR) tools. As we know that any PR task ask for feature extraction & selection. For this a novice binary to real (BTR) technique [1] has been discussed in section 2 & then relevant statistical features are selected. These statistical features form the basis for identification, which has been described in section 3. The said identification problem has been tackled with the help of existing classifiers described in section 4. The graphical display of high dimensional data in two dimensions is explained in section 5 .We has also attempted

to identify the languages from cipher bit stream using these techniques. Experimentation carried out and results have been discussed in detail in section 6. Section 7 concludes with our findings & limitations.

2. CONVERSION OF BINARY TO REAL SEQUENCE (BTR)

Each binary sequences $Y = y_1 y_2 .....y_n$ is converted to real valued sequence $X = x_1 x_2 .....x_n$ by following algorithm illustrated in the form of flow chart



Various statistical features are extracted from this resultant real valued sequence which are explained in the following section

3. FEATURE EXTRACTION

The simulation of classifier demands discritized version of real valued sequence generated in section 2, hence following ten features are extracted from these sequences, say, $F = (F_1, F_2, ...F_{10})$

(i) Mean

The average of real values in given sequence X is

$$F_1 = \frac{1}{n}\sum_{i=1}^{n} x_i$$

(ii) ***Standard deviation***

The variation around mean is studied by

$$F_2 = \sqrt{Var(x_1, x_2, ..., x_n)} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - F_1)^2}$$

The larger value of F_2 signifies more deviation of data from its mean.

(iii) Skewness

The skewness characterizes the degree of asymmetry of distribution around its mean

$$F_3 = Skew(x_1, x_2, ..., x_n) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{x_i - F_1}{F_2}\right]^3$$

A positive value of skewness signifies a distribution with an asymmetric tail towards more positive x. A negative value of skewness signifies a distribution whose tail extends outwards more negative of x.

(iv) Kurtosis

The kurtosis measures the relative peaked ness or flatness of a distribution relative to a normal distribution

$$F_4 = Kurt(x_1, x_2, ..., x_n) = \left[\frac{1}{n}\sum_{i=1}^{n}\left[\frac{x_i - F_1}{F_2}\right]^4\right] - 3$$

In case kurtosis is positive the distribution is a peaked one. The negative kurtosis signifies the flatness in distribution

(v) Entropy

Entropy is the measure of information contained in a pattern. If $X = (x_1 x_2 ..... x_n)$ is the pattern whose entropy is to be calculated and if $t_1 t_2 .... t_{n_1}$ are the distinct values in vector 'X' which has probability $p_1 p_2 ..... p_{n_1}$ where $n_1 < n$, then Entropy of distribution is calculated by

$$F_5 = H(X) = -\sum_{i=1}^{n_1} p_i \ln p_i$$

The value of H lies between 0 and $\ln n_1$. It is zero only when one of the p_i's is one and all others are zero.

(vi) Distinct Values in sequence

The number of distinct values in the real valued sequence X are calculated and taken as a feature

$$F_6 = \#distinct\ values\ in\ sequence\ 'X'$$

(vii) Lowest and Highest magnitude of sequence

$$F_7 = Minimum\ value\ of\ \ sequence\ 'X'$$

$$F_8 = Maximum\ value\ of\ \ sequence\ 'X'$$

(viii) Autocorrelation

Correlation is defined between two different but similar data by comparing them both directly superposed and with one of them shifted right or left [3].

The discrete correlation of two sampled functions g_k and h_k, with a period N, is defined by $Corr(g,h)_\tau \equiv \sum_{k=0}^{N-1} g_{\tau+k}\, h_k$

The correlation function, $Corr(g,h)_\tau$, is a function of τ, which is called the *lag*. The correlation will be large at some value of τ if the first function is shifted to the right of the second (positive lag) i.e. if the function g_k is a close copy of h_k. Similarly, the correlation will be large for some negative values of τ if the first function leads the second i.e. is shifted to the left of the second (negative lag). The relation that holds when two functions are interchanged is

$$Corr(g,h)_\tau = Corr(h,g)_{-\tau}$$

The discrete auto-correlation of a sampled function g_τ is just the discrete correlation of the function with itself. Obviously this is symmetric with respect to positive and negative lags.

To obtain the relevant features, several experimentation were conducted at various lags and leads to finally arrive at the conclusion, that the lags at $\left(\frac{n+2}{4}\right)$ $i.e. F_9$ and $\left(\frac{3n+2}{4}\right)$ $i.e. F_{10}$, gave the most discriminatory values. Hence they become candidate for the feature space F_i.

Each pattern in the form of binary sequence is converted to ten dimensional discriminant vectors, $Z = (F_1, F_2, F_3, F_4, F_5, F_5, F_6, F_7, F_8, F_9, F_{10})$. These F_i's are input patterns to various classifiers for the purpose of identification. These ten features, thus extracted from real sequences vary quite widely in their numeric values. As a result all of them need to be normalized in a fixed interval, say 0 to 1. The normalization of attributes is explained below.

3.1 NORMALIZATION OF ATTRIBUTES

When the attributes are numeric, it is easy to define a distance function. Among many choices, Euclidean distance is most widely used measure to find the closeness between two patterns in pattern-based learning scenario. The distance between a pattern with attribute values Y=$(a_1^1, a_2^1, \ldots, a_k^1)$ (where k is the number of attributes) and one with values Z=$a_1^2, a_2^2, \ldots, a_k^2$ is defines as

$$d(y,z) = \sqrt{(a_1^1 - a_1^2)^2 + (a_2^1 - a_2^2)^2 + \ldots + (a_k^1 - a_k^2)^2}$$

Since different attributes are measured on different scales, so if Euclidean distance formula is used directly, the effect of some attributes might be completely dwarfed by others that have larger scales of measurement. Consequently it is usual to normalize all attribute values to lie between 0 and 1. If there are n patterns in a class each of dimension k then the actual feature value $V(i,j)$ $, i = 1,\ldots n$ & $j = 1,\ldots,k$ are normalized by finding

$$\left.\begin{aligned} t_1 &= Min(V(i,j)) \\ t_2 &= Max(V(i,j)) \end{aligned}\right\} \quad i = 1,\ldots n$$

$$Normalized\ V(i,j) = \frac{V(i,j) - t_1}{t_2 - t_1}$$

We have done experimentation with original feature measurements as well as the normalized feature measurements. The results with normalized data improve considerably and hence they are fed to various classifiers, described in the following section.

4. CLASSIFIERS IMPLEMENTED

The input patterns to the classifiers are ten dimensional data of various languages. To see the success rate, various learning set and test set of each class are taken. The decision rules are formed with the help of learning sets and the same rule is used for test set classification. Linear statistical classifier and minimum distance classifier [4, 5] based on decision theoretic approach and Neural Networks have been used. Classifications with respect to linear statistical classifier and minimum distance classifier have been performed. The Neural Networks approach for language classification is described below.

4.1 NEURAL NETWORKS

A feed forward [6] Multi Layer Perceptron (MLP) [7] model has been chosen in an attempt to classify the plain languages and subsequently their encrypted versions. The MLP is trained using the back propagation algorithm at various learning rates η. A momentum term $\propto$ is also added in this algorithm to speed up the convergence of the network and to determine the effect of past weight changes on the direction of current weight movement [8]. The cross validation technique that will perform testing in various permuted form of data is described below

4.1.1 CROSS VALIDATION

When a limited amount of data is available, then the learning algorithm reserves certain amount of patterns for testing and remainder for training the classifier. It is common to hold one-third of the data out for testing and the remaining two-thirds for training. Here a check is to be made that each of the classes in the full dataset should be represented in about the right proportion in the training and testing sets. If by chance all examples within a certain class were missed out of the training set, no classifier would hardly learn from the data to perform well on the examples of that class, and the situation will be aggravated by the fact that the class would necessarily be over represented in the test set since none of its patterns made it to the training set. So it must be ensured that a random sampling is done in such a way as to guarantee that each class is properly represented in both training and test sets. But this method provides only a primitive safeguard against uneven representation in training and test sets.

A more general way to mitigate any bias caused by the particular sample chosen is to repeat the whole process, training and testing, several times with different random samples. In each iteration a certain proportion, say two-thirds of the data is randomly selected for training and the remainder used for testing. The error or misclassification rates on different iterations are averaged out to yield an overall error rate.

In cross validation, fixed number of folds or partitions of data are decided. In our case the data is split into three approximately equal partitions, each in turn is used for testing while the remainder is used for training. That is, use two-thirds for training and one third for testing, and repeat the process three times so that in the end, every pattern has been used exactly once for testing. This process is called threefold cross validation [9].

We have used neural network with back-propagation algorithm to test the error rate for classification of encrypted version of languages. But generally this algorithm gives different error rates every time we instigate the network (with same or different

configurations). So we chose three different configurations of the network (as in section 6(ii)) and repeated cross validation three times i.e. three threefold cross validation and average the results obtained. Obviously it involves invoking the algorithm nine times on datasets that are all two-third the size of the original one.
The pictorial visualization of any high dimensional data is simplified by functional approximation approach described below

5. FUNCTIONAL APPROXIMATION APPROACH

The functional approximation approach [10] has provided a two-dimensional graphical display of high dimensional data. We have dealt with the transformation that will map higher dimensional data into a function of a single variable. This function is then plotted against the variable presenting a two-dimensional but functional view of the original data. We have already seen that all patterns of various languages are converted to ten dimensional data. Let the data be represented by

$$F = \begin{bmatrix} F_1 \\ F_2 \\ - \\ - \\ F_n \end{bmatrix} = [F_1, F_2, \ldots F_n]^T$$

We define the functional transformation for a given F as

$$f_F(t) = \frac{1}{\sqrt{2}} F_1 + F_2 \sin t + F_3 \cos t + F_4 \sin 2t + F_5 \cos 2t + \ldots \begin{cases} F_n \sin \frac{n}{2} & \text{if } n \text{ is even} \\ F_n \cos \frac{n-1}{2} & \text{if } n \text{ is odd} \end{cases}, -\pi < t < \pi \quad \text{---- (1)}$$

This $f_F(t)$ for each F is plotted against t for –3.14 ≤ t ≤ 3.14 at same constant interval. The functional mapping in (1) is linear, preserves the mean vector, distances, and variances. The results of this mapping are shown in fig 4, 5 and 6.
From figure 4 we can find out the deterministic region in the case of plain languages, which signifies that the choice of attributes are proper and they are discriminatory in nature. Overlapping can be observed in the case of crypts (Fig 5 and 6), which means that although the choice of features is good but crypt need some more highly discriminatory features to classify among themselves.

6. RESULTS AND DISCUSSIONS

49 romanized messages of Punjabi, 35 romanized messages of Hindi, and several messages of English, 49 in our case, were converted to bit stream using 8 bit ASCII code. These plain text were encrypted through DES (crypt1) and Geffe Generator (crypt2) crypto algorithms. The BTR technique explained in section 2 was used to convert these sequences into real valued sequence. The ten features described in section 3 were calculated for the three plain languages and their respective encrypted versions. These features were then given to various classifiers explained in section 5. The functional approximation approach was also incorporated to get insight into graphical representation of plain and encrypted data of English, Hindi and Punjabi.

The experimentation and studies on the plain and encrypted languages using various approached are shown below.

(i) Using Linear Statistical Classifier & Minimum Distance Classifier

In this method we initially started with first 5 features. The results were found to be extremely good in pair-wise classification of English, Hindi and English, Punjabi in comparison to Hindi, Punjabi. The results on these features for encrypted data were also not satisfactory. This anomaly leads to the further inclusion of more features so as to make the feature space as ten dimensional. After this amendment both the results improved. One more observation was that order of magnitude of features F_9 & F_{10} was quite high in comparison to the other features. So when a linear discriminant classifier or any distance metric was to be incorporated, these two features always over-shadow the effect of rest of them and we could not get desired results. But when normalization (refer section 4) was attempted on features individually and independently the performance of the classifier enhanced. Furthermore, the bar chart representation of percentage success using 10 normalized features in linear statistical classifier is shown in fig 1, 2, and 3.

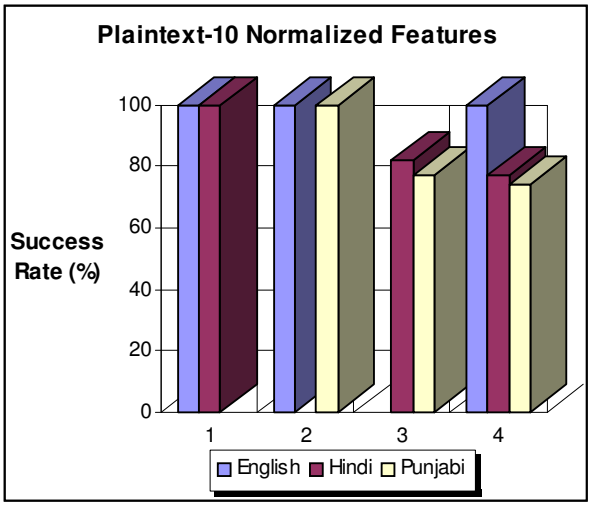


Fig 1

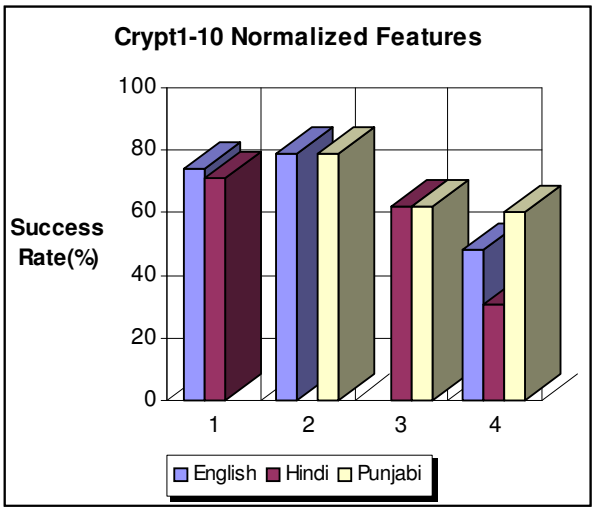


Fig 2

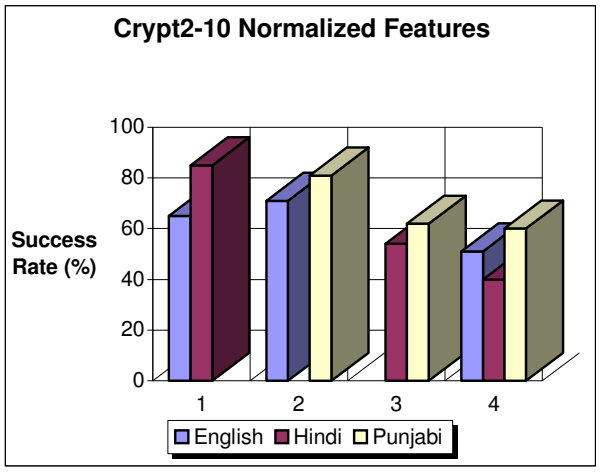


Fig 3

(ii) Using Neural Network

The neural network was trained using feed forward MLP, with one hidden layer. The input neuron corresponds to the number of features representing the pattern and the output neuron corresponds to the desired output. Hence these two are fixed. The only variable is the number of neurons in the hidden layer, and to note the effect while the network learns on the training the data. Three different configurations of network were studies viz. 10-3-1 (10 neurons in the input layer, 3 in hidden layer and 1 in output layer), 10-8-1 and 10-15-1. Each of these three configurations was repeated using threefold cross-validation process and the error or the misclassification rate is averaged out. The network was trained with learning rate of 0.15 and momentum rate 0.35 for the input and hidden layers. The results for plain languages are shown in table 1.

File Type	**Error Rate (%age) (three three-fold cross validation) ≈**
Normalized English, Hindi plain	2
Normalized English, Punjabi plain	1.5
Normalized Hindi, Punjabi plain	18

Table 1

The experimentation on encrypted version of languages is in progress.

(iii)Using Functional Approximation Approach

In fig 4, the plain English, Hindi and Punjabi can be easily discriminated. In fig 5 and 6, the encrypted versions of languages shows overlapping at various places which might improve by using some non linear classifier.

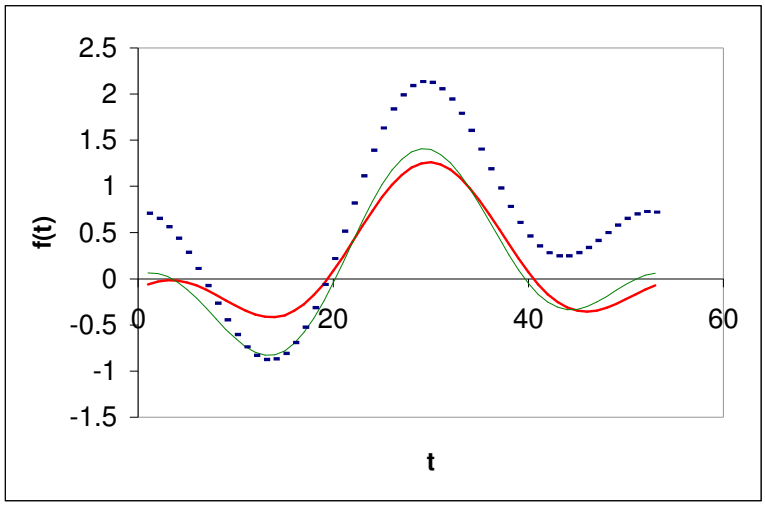


Fig 4. Plain

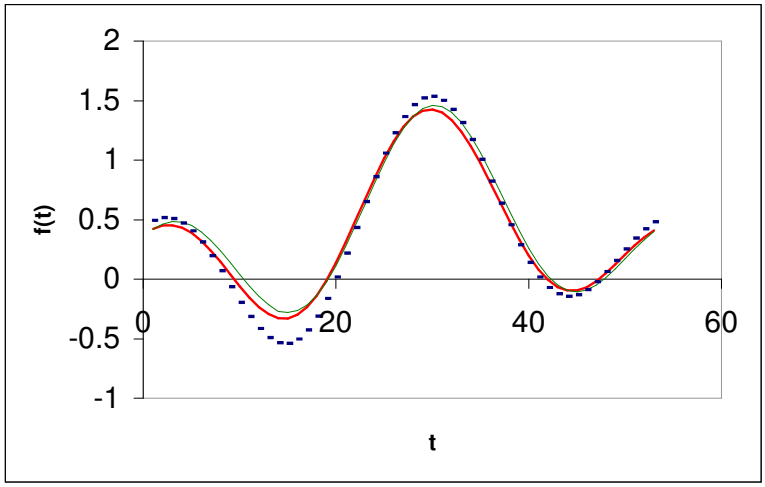


Fig 5. Crypt1

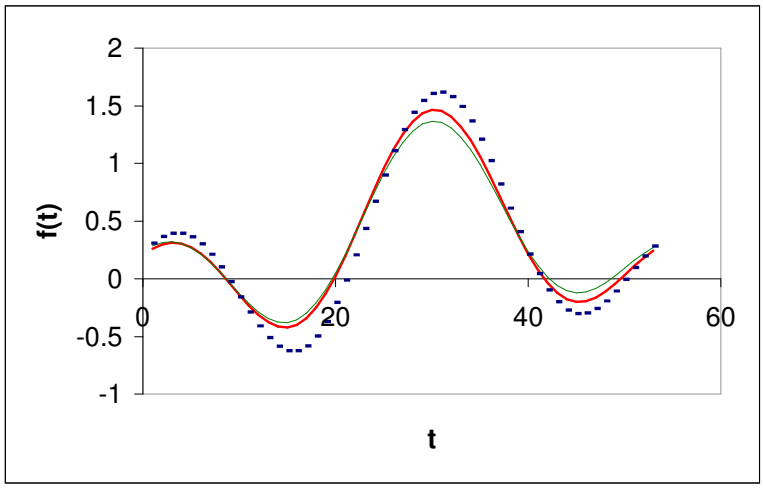


Fig 6. Crypt2

English Hindi Punjabi

With all these results in hand we can say that that the features we choose with intent to identify different plain languages are outstanding. The results on encrypted languages need further improvement.

7 CONCLUSIONS

From the above result and discussion it has been observed that discrimination among bit stream of plain languages is not a problem. The problem arises when we attempt to recognize the languages from the cipher bit stream. When we moved from five features to ten features the results got improved significantly. It clearly shows that there is a need of searching new features, which can be utilized for getting performance by these classifiers. We hope that new approaches of classification can be formulated to arrive at exhilarating performance in terms of classification among encrypted bit stream of various languages. In the present paper we have studied only three languages, the same can be extended to study the other romanized language to maintain the security of the nation.

ACKNOWLEDGEMENT

We express our deep gratitude to **Prof. C.E. Veni Madhavan**, Director SAG, for his continuous encouragement and support. The authors are highly thankful to **Dr. P. K. Saxena**, Sc 'G' and his group members to cooperate us in procuring the romanized data.

REFERENCES

[1] Khan, S.S, Verma Neelam, "Extracting Statistical Features to identify Plain and crypt sequences", Proceedings of National Conference on Sensor Technology, 2002

[2] Stalling W., "Cryptography and Network Security", Prentice Hall 1999

[3] [Press, W.H., Vellerling, W.T., Teukolsky, S.A., Flannery, B.P., "Numerical Recipes in C", Second Edition, 1995

[4] Tou, J.T, and Gonzalez, R.C. "Pattern Recognition Principles", Addison Wesley (1981)

[5] Bow, S.T., "Pattern Recognition", Marcel Dekker, Inc, 1988

[6] Hertz J, Krogh A, Palmer RG, "Introduction to the Theory of Neural Computation", Addison Wesley, Reading, MA, 1991

[7] Raudys, S., "Statistical and Neural Classifiers: An integrated Approach to Design", Springer-Verlag London Ltd. 2001

[8] Tzanakou E.M., "Supervised and Unsupervised Pattern Recognition: Feature Extraction and Computational Intelligence", by CRC Press LLL, 2000

[9] Witten, I.H., Frank, E., "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann Publishers, 2000

[10] Yi-Tzuu Chien, "Interactive Pattern Recognition", Marcel –Dekker Inc, 1978