Defence Science Journal, Vol. 62, No. 1, January 2012, pp. 38-45, DOI: 10.14429/dsj.62.1440

Classification Models for Symmetric Key Cryptosystem Identification

Shri Kant

Joint Cipher Bureau, Delhi – 110 054, India
E-mail: shrikant.ojha@gmail.com

ABSTRACT

The present paper deals with the basic principle and theory behind prevalent classification models and their judicious application for symmetric key cryptosystem identification. These techniques have been implemented and verified on varieties of known and simulated data sets. After establishing the techniques the problems of cryptosystem identification have been addressed.

Keywords: Classification, supervised pattern recognition, feature extraction, decision surfaces, discriminant functions

1. INTRODUCTION

First, we will define the two interrelated terms discrimination and classification as discussed in Kendall[1]. In discrimination we are given samples from *k*-populations. Each member of these populations has been described with *p*-number of variables. Our job is to construct a method for assigning a new member to the correct population on the basis of the *p*-variables. Whereas, in classification we are given sample of members this may or may not emanate from the same population. We are required to set up a method of deciding whether the members fall into groups, and if so, to delineate the groups. In due course of time both the term got united and most of the research continued in the name of classification.

New methods and models of theoretical classification research have attracted the researchers from the varied areas like anthropology, archeology, astronomy, biology, business, chemistry, computer science, economics, engineering, geography, marketing, geology, information theory, cryptology, medicine, political science, psychology, soil science, military science, and other areas indicate the importance of existing classification tools and the need for further development.

Lance and Williams[2,3] have defined different genus of classification problem (Fig. 1), an exclusive classification is a partition of the set of objects. Each object belongs to exactly one subset. Non-exclusive or overlapping classification can assign an object to several classes. For example, grouping of people by age or sex is exclusive, whereas grouping of people by disease category is non-exclusive, because a person can have several diseases.

Intrinsic classification (unsupervised learning or clustering) uses only the proximity matrix to perform the classification and no category labels denoting a priori partition of the objects are used. In extrinsic classification (supervised learning) category labels on the objects as well as the proximity matrix both are required. The problem is then to establish a discriminating surface that separates the objects according to the category.

In hierarchical clustering, the objects are ordered in such a way that the individual objects in the same cluster at any level remain together at all higher levels[4,5]. The process of hierarchy construction becomes computationally infeasible as the data size increases. And, hence for large data size partitional or non-hierarchical clustering is well suited because the objects are taken serially.

The extrinsic or supervised classification, i.e. statistical pattern recognition are of two type as parametric learning and non-parametric learning. In parametric approach to supervised learning, we require the underlying class conditional pattern distributions (in both form and parameter). In practice the following problem may arise.

- It is difficult to determine a specific form (Gaussian or uniform) for the distribution.
- The form chosen does not fit into one of the 'estimated' formulations.

And hence we resort to non-parametric learning techniques with the help of a variable labeled training set for each class. The approaches are:

- Estimation of density function $p(x/w_i)$ directly
- Directly estimate $p(w_j/x)$

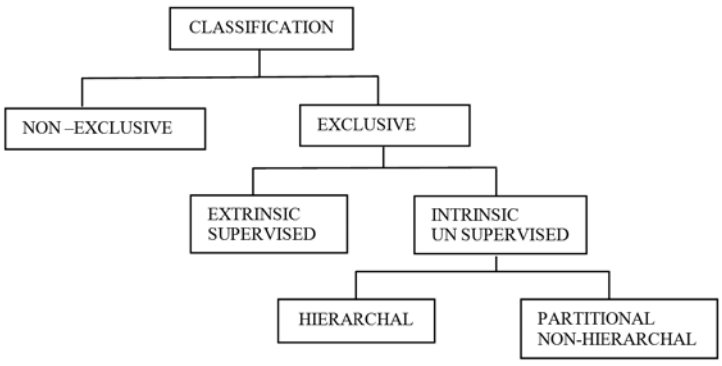


Figure 1. Genus of classification.

Received 15 December 2011, online published 23 January 2012

- Transform the feature space.

The focus here is to develop generalised extrinsic classification models, so that they can be tailored for any application.

2. STAGES OF CLASSIFIER

The various stages involved in classifier design are depicted in Fig. 2. Sensor or data generator is supposed to provide input to the feature generation algorithms. Then, we have to address the four basic questions. How to generate the features in measurement space? It is problem dependent, and it depends heavily on the domain knowledge of the designer and to be addressed at the feature generation stage. Algorithms have been developed for generating the features from cipher text[6]. How many and which features are to be used? The problem here is to get rid of redundant features and also to save the design from the curse of dimensionality. At the feature selection/extraction stage these issues are to be addressed. At classifier design stage how to find out a function or rule, which maps the entire labeled pattern, made up of selected feature vector to the right class? Classification models developed will be discussed briefly. The obtained decision function should also be able to recognise the new pattern of the learned classes. How can one assess the performance of the designed classifier? The simplest way for measuring the performance is to have minimum classification error rate during validation and testing. The crypto-system identification results have been discussed after each of the model.

It is apparent from Fig. 2. that the different stages are not independent. The stages are interrelated and, depending on the results, one may go back to redesign earlier stages in order to improve the overall performance. For the purpose of crypto-system identification we have designed a general classification model depicted in Fig. 3. This model is used for the designing of classifier for the identification of crypto-system at macro- and micro-level.

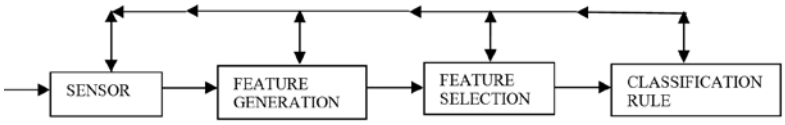


Figure 2. Stages of a classification system.

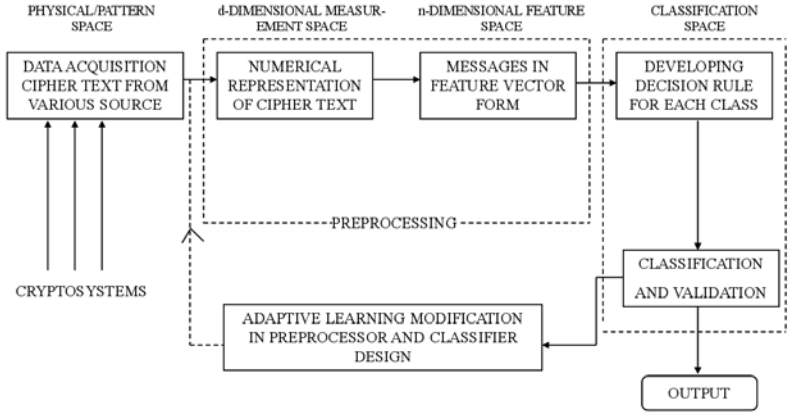


Figure 3. Block diagram of a crypto-system identifier.

3. THE CURSE OF DIMENSIONALITY

The performance of any classifier depends on the interrelationship between sample sizes (n), the number of features (d) and the classifier complexity. In general, the number of training data points are found to be an exponential function of the 'features dimension'. This phenomenon is termed as 'curse of dimensionality'.

- We also know, that the probability of misclassification of a decision rule does not increase as the number of features increases as long as the class-conditional densities are completely defined.
- It has been observed in practice that the added features may actually degrade the performance of a classifier if the number of training samples that are used to design the classifier are small in relation to the number of features.

Then what should be the size of training set? Literature[8-10] is full of such discussion. In general, it is advised to select small number of salient features. Accepted norm is ((n/d)>10): i.e. number of training samples should be 10-times more than the number of features.

3.1 Dimensionality Reduction and Feature Selection

Feature selection or reduction is a process wherein we focus on; given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible the class discriminatory information? In other words, the objective of feature selection is to reduce the dimensionality of the measurement space to a space where salient features necessary for the recognition process are retained so that classification can be implemented on a vastly reduced feature space.

It must be emphasised that feature selection is very crucial. If we select features with little discrimination power, the subsequent design of a classifier would lead to poor performance. On the other hand, if information-rich features are selected, the design of a classifier would give good result. So, we should aim to select features leading to large between-class distance and small within-class variance in the feature vector space. This means that features should take distant values in the different classes and closely located values in the same class. To this end various approaches can be adopted. One has to examine the features individually and discard those with little discriminatory capability. A better alternative is to examine them in combinations. Sometimes the application of a linear or non-linear transformation to a feature vector may lead to a new one with better discriminatory properties. All these paths will be our touring directions in the following subsections.

3.2 Parametric Feature Selection Methods

Statistical inference is to exploit experimental data to decide whether a certain ideal model can be assumed to be the one representing an actual phenomenon. Here, we estimate the parameter of a population by using various samples of this population. When we make assumptions about some aspects of distribution of the sampled population, e.g.

we may assume that the population is normally distributed then the tests like test of hypothesis concerning population mean or variance or equality of means etc. are the test of hypothesis about parameter of known population types. Such tests are known as classical tests or standard tests or parametric tests.

In these algorithms, we should have knowledge about the distribution function of the pattern classes. We have applied single population't' test, two-population 't' test, and ANOVA techniques for feature selection[11].

3.3 Non-parametric Feature Selection

We may come across situations where some or all of the assumptions required in parametric tests are not met, in that situation we will use those statistical tests that are not based on assumption about distribution of the sampled population. These tests are known as non-parametric or distribution free tests. Non-parametric tests are based on the order statistics theory. Non-parametric procedure should be used when any of the following is true.

- The data are measured on a nominal scale.
- The data are measured on an ordinal scale.
- The assumptions required for the validity of parametric procedure are not met.
- The shape of the distribution from which the sample is drawn is unknown
- The sample size is small.

To overcome these difficulties nonparametric tests viz., Sign test, Wilcoxon signed rank test, Mann Whitney U test, etc. are used for feature selection[9].

3.4 Feature Selection through PCA

In this algorithm, the correlated features are transformed to uncorrelated features in such a way that maximum information is retained in few uncorrelated transformed features. This method of reducing dimensionality has given good results in many classification techniques. So, the goal is to get principal component Z_1 , Z_2, … , Z_P from the original variable F_1, F_2, …, F_P, in such a way that Z_1, the first principal component has the highest variance, Z_2, to the second principal component has next highest variance, and so on[11,12].

4. CLASSIFIER DESIGN

In the quest of cryptosystem identification, the relevant classification models from statistical pattern recognition, statistical learning theory, artificial neural network and support vector machine have been developed. The models have been developed in such a way that once the observation of any problem of identification converted into feature vector form is available, then they can be fed to these models for their respective classification. And hence they are termed as generalised classification models[13,14]. Some models, which got only mention in the literature, have been formulated and developed specially for cryptosystems identification viz. minimum distance classifier (MDC) or linear statistical classifier[15,16] has been found very effective in case of identification between classical cryptosystems. The piecewise linear classifier (PLC) and maximum likelihood classifier (MLC)[17,18] are used for the study of electromechanical and electronic ciphers. Decision fusion approach has been proposed and developed for specific applications[19]. Functional approximation approach has been developed with different basis functions to estimate the densities of unknown distributions.

4.1 Functional Approximation-based Classifier

In the above mentioned classification models the underlying assumption is normality for which probability density function (PDF) is known and accordingly decision functions are determined for classification. The parameters of a probability density function can be easily determined if its form is known. But in many cases we don't know the form of the PDF, so we have to estimate the PDF directly from the data. In functional approximation method we find out the approximate probability density function of various data sets of different classes to be learnt.

Let, $\hat{p}(x)$ represents an estimate of $p(x)$, where $p(x)$ i.e. $p(x/\omega_i)$ is Likelihood function of class ω_i .We wish this estimate should minimise the mean square error function, defined as

$$E = \int_x w(x)\left[p(x) - \hat{p}(x)\right]^2 dx ,$$

where, $w(\boldsymbol{x})$ is a weight function. Let us expand the estimate $\hat{p}(x)$ in the series

$$\hat{p}(x) = \sum_{j=1}^{m} c_j \phi_j(x)$$

where the c_j are coefficients to be determined and $\{\varphi_j(\boldsymbol{x})\}$ are a set of specified basis function. It could be derived from any special functions viz., Hermite, Legendre, Laguerre or Bessel functions, etc. Coefficients c_j are computed by following expression

$$c_j = \frac{1}{N}\sum_{i=1}^{N} \phi_j(x_i), \quad j = 1, 2, ..., m$$

Once the coefficients have been determined, the probability density function $\hat{p}(x)$ is formed and the error function E can be computed. The successful application of functional approximation method requires two basic considerations:

(i) The quality of approximation for a chosen set of basis functions depends on the number (m) of terms taken.

(ii) Another important consideration is the choice of basis functions.

Suppose we have n-dimensional N data sets for m classes. We estimate PDF $p(x/\omega_i)$ as described above for each class i, where i=1, 2,...,m. We know the priory probabilities $p(\omega_i)$ for each class. Then we form decision functions $d_i(x)$ as

$$d_i(x) = p(x/\omega_i) \,.\, p(\omega_i)$$

We compute the reference value for each class from the above expression by taking x as the mean of the learning data sets (n-dimensional) for each class. Then for testing we compute $d(x)$ by the above expression and take absolute distance from reference values for each class and assign to that class for which it has the minimum distance.

4.2 Experimental Setup for Cryptosystem Identification

In general the classification models are developed with the labeled patterns available and validated also with the labeled pattern by removing the label of pattern to be used during validation. Once the confidence achieved, the parameters and the prototypes for each class is stored for future reference. In our case since the success rate during training or validation is not very high hence we train the model on very large randomly selected data set and go for blind testing. By blind testing, we mean that cipher text generated for testing is on different set of keys which are not used during training and validation and we look for percentage of success on these test messages.

4.2.1 Experiment 1

Two sets of data file each consists of 10,000 ciphered messages have been generated from block and stream ciphers. Two feature selection techniques 'two-population 't' test' and 'projection pursuit (linear functional mapping)' are used to get 10,000 pattern vectors from each class. The first 6,000 messages from each class has been used for training the three classifiers viz., minimum distance classifier (MDC), piecewise linear classifier (PLC), maximum likelihood classifier (MLC) and computed the weights corresponding to the selected features. The remaining 4,000 patterns from each class have been used for testing. It has been observed that the classification accuracy is up to 82 per cent in case of learning and up to 75 per cent in case of testing. The MLC is outperforming the other two classifiers.

4.2.2 Experiment 2

The newly proposed classification model, functional approximation based classifier has been first tested on well known ionosphere data [UCI data repository http://www.sgi.com/tech/mlc/db/], comprising of two classes. There are 125 data sets characterised with 34 features. First we trained the classifier by taking 25, 50, 75, and 100 patterns from each class and computed weights corresponding to the features. Then, the remaining 100, 75, 50, and 25 patterns are exposed for testing. The Hermite and Legendre polynomials are used for estimation of probability densities. The results are summarised in Table 1.

Table 1. Classification performance using hermite and legendre polynomial as basis function.

Data sets for learning	Hermite polynomial % success		Legendre polynomial % success	
	Class I	Class II	Class I	Class II
25	99.51	96.00	100.00	97.16
50	100.00	96.84	100.00	97.91
75	100.00	97.70	100.00	98.33
100	100.00	100.00	100.00	100.00

4.2.3 Experiment 3

The features from the data as described in experiment 1 are extracted with 'Wilcoxon signed rank test' and the pattern vector obtained have been exposed to the functional approximation-based classifier for training as well as for testing. It has been observed that the FABM is providing consistent results up to 75 per cent in terms of classification accuracy.

4.3 Classification Models based on Artificial Neural Network[20]

A neural network is a massively parallel distributed processing made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

(a) Knowledge is acquired by the network from its environment through a learning process.
(b) Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

For training the ANN models datasets were generated using each of the cipher systems. Various 12.8 K bytes of plaintext messages were encrypted using 1,000 randomly generated keys. The size of each dataset is about 12.8 million cipher characters of which 10.24 million cipher characters were used for training and the remaining were used for testing in each cipher system.

4.3.1 Data Transformation Technique[21]

Large collection of cipher texts was generated from different sets of plaintext messages (12.8 K bytes) using different encryption algorithms. Different sets of plaintext messages and keys were used for each algorithm. For crypto system identification, the ASCII values of the cipher texts were considered. A group of 32 ASCII values were considered as one input pattern. The message of 12.8 K Bytes resulted in 400 patterns each containing 32 ASCII values of the corresponding cipher characters. Various neural network models were trained using this data but the testing accuracy was low. This is due to the fact that the cipher text characters are a pseudorandom sequence. Various standard data transformation techniques like $1/x$, $\sqrt{x}$, $\log x$ was applied on the cipher text data for improving the testing accuracy but the results were not encouraging. Summation of 20 patterns was taken and then a first order difference of the features in each of these summed up pattern was taken to make the series stationary. The resultant patterns were fed then to the neural network. Classification on the first order-differenced series was carried out on various neural network models with different architectures. It was observed that there was some improvement in the results.

To give greater importance to the smaller values of first order difference, the data was further transformed using the exponential transformation of the negative of first order difference. Then cross correlation values between features of the transformed data (30-dimensional) of each cryptosystem alongwith 16 reduced features (using principal component analysis) of the summed up patterns was fed to various neural network models.

4.3.2 Resilient Back Propagation

The purpose of the resilient back propagation (RPROP) training[22] algorithm is to eliminate the harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative is used to determine the direction of the weight

update; the magnitude of the derivative has no effect on the weight update. The size of the weight change is determined by a separate update value. The update value for each weight and bias is increased by a factor Δ whenever the derivative of the performance function with respect to that weight has the same sign for two successive iterations. The update value is decreased by a factor Δ whenever the derivative with respect to weight changes sign from the previous iteration. If the derivative is zero, then the update value remains the same. Whenever the weights are oscillating the weight change will be reduced. If the weight continues to change in the same direction for several iterations, then the magnitude of the weight change will be increased.

Every time the partial derivative of the corresponding weight change w_{ij} changes its sign, it indicates that the last update was too big and the algorithm has jumped over a local minimum, the update value Δ_{ij} is decreased by a factor of η^-

$$\Delta w_{ij} \begin{cases} -\Delta_{ij}^{(t)}, \; if \; \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, \; if \; \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, \; else \end{cases}$$

$$\Delta w_{ij} \begin{cases} \eta^+ \times \Delta_{ij}^{(t-1)}, \; if \; \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^- \times \Delta_{ij}^{(t-1)}, \; if \; \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)}, \; else \end{cases}$$

where, $0 < \eta^- < 1 < \eta^+$

If the derivative retains its sign, the update value is slightly increased in order to accelerate convergence in shallow regions. However there is one exception: If the partial derivatives change sign, i.e. the previous step is too large and the minimum is missed, the previous weight change is reverted.

$$\Delta w_{ij}^{(t)} = -\Delta w_{ij}^{(t-1)}, \; if \; \frac{\partial E^{(t-1)}}{\partial w_{ij}} \frac{\partial E^{(t)}}{\partial w_{ij}} < 0$$

Results of RPROP Classification Tool

Letter recognition data discrimination [UCI data repository http://www.sgi.com/tech/mlc/db/]: We have experimented the above RPROP ANN model to discriminate between letter data. The feature vector of letter data has 16 feature and 750 patterns for each character. The letter may be either hand written or typed by any machine, which converted it to 16 dimension patterns. We have used 500 patterns for learning and rest 250 patterns for testing, so that the discrimination between most difficult classes viz. (X and Y), (O and D), (U and V) and (O and Q) could be ascertained. The result quoted in literature is not exceeding 85 per cent to 95 per cent but with the help of RPROP we got the classification accuracy between 97.2 per cent to 98.8 per cent.

We trained the model using 1,600 patterns with learning rate 0.2 and accuracy 0.005 to discriminate between A5/1(stream cipher system) and AES (block cipher system). Then tested for 12,400 patterns, average results are as follows:

Self Classification Results			
Training data class	**Patterns**	**A5**	**AES**
A5	1600	96.8	3.2
AES	1600	2.0	98.0
Testing Results			
Test data class	**Patterns**	**A5**	**AES**
A5	12400	53.41	46.59
AES	12400	26.66	73.34

4.3.3 Self Organising Map

Self organising maps (SOM) are a type of unsupervised learning. The goal is to discover some underlying structure of the data and used for clustering and classification. The architecture of SOM is as shown in the Fig 4. It has one input layer and one output layer. The number of neuron in input layer is equal to the dimension of a pattern vector. The number of neurons in output layer is equal to the number of clusters. It discovers the inter relationship of the pattern vectors and group the patterns which has similar semantic meaning. Each output neuron j is connected to each input neuron i by a link which has some weight w_{ij}. Weight vectors $w_j = \{ w_m(0) ; \; i \le m \le d$ (dimension of pattern vector) $\}$ where $j = 1,2,3,\ldots,n$ (no of neurons) represent weight of individual link from input layer to output layer.

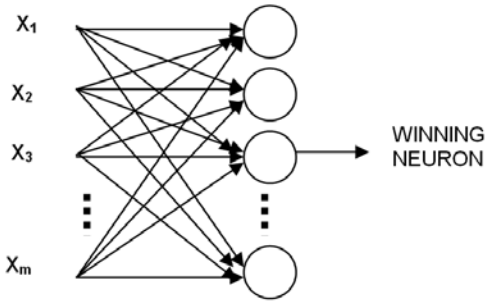


Figure 4. Self organising map network architecture.

Algorithm

(i) Initialise l weight vectors w_j (0), where j=1, 2, 3,…, l by picking l patterns randomly from input space.

(ii) Draw the sample pattern from input space with a certain probability; the vector x represents the activation pattern that is applied to the network.

(iii) Find the best matching (winning) neuron $i(x)$ at time step n by using the minimum distance Euclidean criteria:

$$i(n) = \arg\min_j \left\| x(k) - w_j \right\|, \; j = 1,2,3,\ldots,n$$

(iv) Adjust the weight vectors of all neurons by using the update formula:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)(x(n) - w_j(n))$$

where, $\eta(n)$ is the learning rate parameter and $h_{j,i(x)}(n)$ is the neighborhood function centered around the winning neuron $i(x)$.

(v) Continue with step 2 until no noticeable changes in the weight vectors are observed.

Experimental Results

Cryptosystem discrimination: We trained the model using 3,200 patterns to discriminate between A5/1 (Stream cipher system) and AES (Block cipher system). Then tested for 16,000 patterns, average results are as follows:

Self classification results			
Training data class	**Patterns**	**A5**	**AES**
A5	3200	94	6
AES	3200	0	100
Testing results			
Test data class	**Patterns**	**A5**	**AES**
A5	16000	55.33	44.67
AES	16000	23.40	74.60

4.3.3 Learning Vector Quantisation

Input space is divided into a number of distinct regions (Vornoi cells) and for each region a reconstruction vector (Vornoi vector) is defined. When the quantiser is presented a new input vector, the region in which the vector lies is first determined, and is then represented by reproduction vector for that region.

Self organising map algorithm provides an approximate method for computing the Voronoi vectors in an unsupervised manner at first stage. The second stage is provided by learning vector quantisation, which provides a mechanism for the final fine tuning of the feature map.

Algorithm: Let $\{w_j\}_{j=0}^{l}$ denote the set of Voronoi vector and let $\{x_i\}_{i=0}^{N}$ denote the set of input vectors.

(i) Suppose that the Vornoi vector w_c is the closest to input vector x_i.

If Class of w_c = Class of x_i

$$w_c(t+1) = w_c(n) + \alpha_n[x_i - w_c(n)]$$

If Class of $w_c \neq$ Class of x_i

$$w_c(t+1) = w_c(n) - \alpha_n[x_i - w_c(n)]$$

(ii) The other Voronoi vectors are not modified.

It is desirable for the learning constant α_n to decrease monotonically with the number of iterations n.

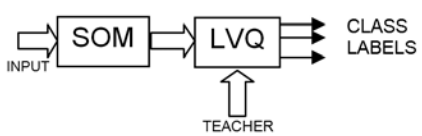


Figure 5. Block diagram of adaptive pattern classification, using self organising map and learning vector quantizer.

Experimental Results

Cryptosystem discrimination: We trained the model using 3200 patterns to discriminate between A5/1(Stream cipher system) and AES(Block cipher system). Then tested for 16,000 patterns, average results are as follows:

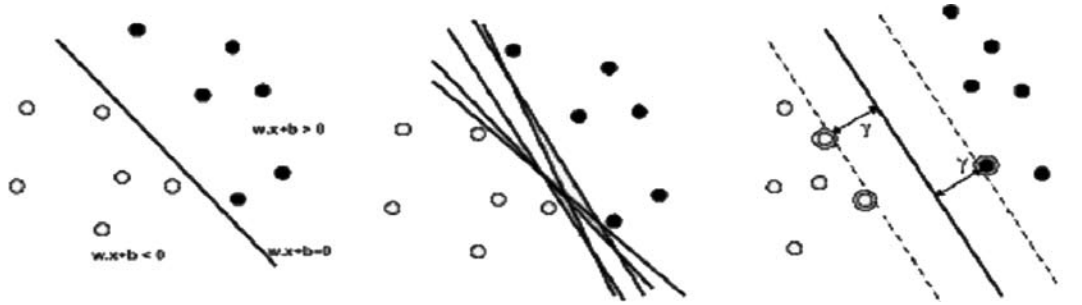
Self classification results			
Training data class	**Patterns**	**A5**	**AES**
A5	1600	95	5
AES	1600	0	100
Testing results			
Test data class	**Patterns**	**A5**	**AES**
A5	16000	52.82	47.18
AES	16000	18.86	81.14

4.4 Support Vector Machines

Support vector machines (SVM) is a relatively new classifier and is based on strong foundations from the broad area of statistical learning theory[23]. Since its inception in early 1990s, it has found applications in a wide range of pattern recognition problems, to name a few: handwritten character recognition, image classification, financial time series prediction, face detection, bioinformatics, biomedical signal analysis, medical diagnostics and data mining. Support vector machines has become, in practice, the classifier of choice of numerous researchers and practitioners for several real-world classification problems. This is because SVM is capable of generalising well (predicting the unseen or unknown samples with a good degree of accuracy) as compared to many traditional classifiers. It offers several advantages which are typically not found in other classifiers:

- Computationally much less intensive (especially in comparison to ANN)
- Performs well in higher dimensional spaces (a factor which limits many efficient classifiers).
- Lack of training data is often not a severe problem
- Based on minimising an estimate of test error rather than the training error (structural risk minimisation)
- Robust with noisy data (noise can severely degrade the performance of ANN)
- Does not suffer as much from the curse of dimensionality and prevents over fitting.

A binary class supervised classification problem is usually formulated in the following way: given n training samples $(< x_i >, y_i)$ where $< x_i >= (x_{i1}, x_{i2}, ..., x_{im})$ is an input feature vector and $y_i \in \{-1, +1\}$ is the target label, the task of the discriminant function or a classifier is to learn the patterns in the training samples in such a way that at a later stage it can predict reliably a y_i for an unknown x_i. SVM is fundamentally developed for such binary classification case and is extendable for multi-class situation. Like other linear classifiers, it attempts to evaluate a linear decision boundary (assuming that the data is linearly separable) or a linear hyperplane between the 2-classes



(a) Linear hyperplane **(b) Various hyperplanes** **(c) Optimal hyperplane**

Figure 6. Evaluation of an optimal hyperplane.

(Fig 6 (a)). Theoretically, when the data is linearly separable, there exist possibly an infinite number of hyperplanes (Fig 6 (b)) which can correctly classify the training data. SVM, unlike other classifiers of its kind, strives to find out an optimal hyperplane (Fig 6 (b)). It is commonly believed that points belonging to the two data classes often lie in such a way that there is always some 'margin' between them. SVM attempts to maximise this margin (2γ in Fig 6 (c)) by considering it as a quadratic programming problem.

4.5 Statistical Learning Theory–Key to Generalisation for Support Vector Machines

Support vector machines (SVM) foundations are strongly rooted in the statistical learning theory (SLT): an area of study which addresses the issue of how to control the generalisation ability of a classifier. Vapnik and Chervonenkis[24] developed this theory for creating classifiers or discriminant functions that can generalise well. Normally, classifiers like ANN try to minimise the training error (called empirical error) but, statistically, there is no guarantee that such a classifier (with zero empirical error) will perform well on the test data. SLT, in contrast, provides an estimate of the test error (called risk):

$$Er \le Ee + c\sqrt{\frac{V(F)}{N}}$$

where, *Er* is the risk, *Ee* is the empirical error, *c* is a constant, *V*(*F*) is the Vapnik-Chervonenkis dimension (VC-dimension), denoting the maximum number of samples that can be shattered by a set of functions, and *N* is the number of training samples. This equation describes the basis of the structural risk minimisation principle. SVM, in practice, is based on the minimisation of this risk estimate which often leads to efficient classifiers.

4.5.1 Kernel Function – The kernel-trick

The concepts above are presented for a linear classification case. These are generalisable to a nonlinear case where a mapping function $\phi(x_i)$ is used to map the input space into a higher dimensional feature space such that the non-linear hyperplane becomes linear (Fig 7). To avoid the increased computational complexity and curse of dimensionality, a kernel-trick or kernel function $K(x_i, x_j)$ is employed which, in essence, computes an equivalent kernel value in the input space such that no explicit mapping is required[25].

A few popular kernels are:

Linear: $K(x_i, x_j) = \langle x_i, x_j \rangle$

Gaussian: $e^{-\gamma \|x_i - x_j\|^2}$

Polynomial: $\left(\gamma \langle x_i, x_j \rangle + a\right)^d$

4.5.2 Experiment and Observation

Due to above mentioned characteristics we have made use of SVM for predicting the key bit of block ciphers and also for identification among block ciphers by studying the features of ciphers generated from these block ciphers[26-28] .

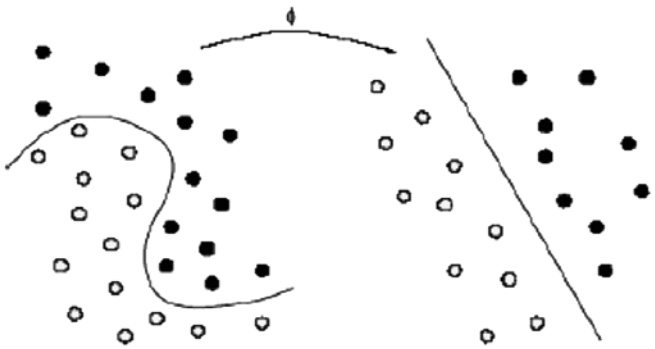


Figure 7. Feature mapping.

5. DISCUSSION AND CONCLUSIONS

(i) The models developed are not limited to the models briefly described in the paper. We have also implemented and carried out experimentation with perceptron algorithm (deterministic), hidden Markov model (HMM), Fisher's discriminant analysis and various other models from artificial neural network like variants of back propagation, radial basis function (RBF), multilayer feed forward network, recurrent neural network (RNN), etc. The success of crypto system identification is in the same range as mentioned in the paper. Concluding views have emerged that the ANN model along with support vector regression may provide better classification accuracy for crypto-system identification.

(ii) Hidden Markov model is a strong and powerful tool and has been used particularly for speech recognition and image analysis. However, the method is quite complicated and the implementation requires numerous computation. There is no evidence of this method having been used in our context before, so there is no existing standard, which makes it difficult to use. We have finally implemented the hidden

Markov model and the results generate some hope of partial success in cryptanalysis.

(iii) A fair amount of development has been carried out in the area of unsupervised learning and one class classification problem using cluster analysis and machine learning techniques. Due to crunch of space no discussion for this is possible in this paper but a very outstanding result has been obtained in the direction on classificatory next bit prediction[29], which may finally leads to solution for symmetric key stream crypto primitives.

REFERENCES

1. Kendall, M.G. Discrimination and classification. *In* Multivariate Analysis, edited by P.R. Krishnaiah, Academic Press, New York, 1966, pp.165-85.
2. Lance, G.N. & Williams, W.T. A general theory of classificatory sorting strategies 1 hierarchical system, *Computer Journal*, 1966, **9**, 373-80.
3. Lance, G.N. & Williams, W.T. A general theory of classificatory sorting strategies 2 clustering systems, *Computer Journal,* 1966, **10**, 271-77.
4. Anderberg, M.R. Cluster analysis for application. Academic Press, New York. 1973.
5. Ball, G.H. & Hall, D.J. A clustering technique for summarising multivariate data. *Behav. Sci.,* 987, **12**, 1153-155.
6. Shri Kant, *et.al.* Technical report on statistical analysis tools, (SAMEEKSHA) No.RD-P1-2003/SAG-09, Report No. SAG/ STAT-EVAL/C/2005-06, 2006.
7. Jain, A.K.; Duin, Robert P.W. & Mao, J. Statistical pattern recognition: A review. IEEE, PAMI, 2000, **22**(1), 4-37.
8. Shri Kant & Verma, Neelam. An effective source recognition algorithm: Extraction of significant binary words. *Pattern Recognition Letters*, 2000, **21**, 981-88.
9. Duda, R.O.; Hart, P.E. & Stork, D.G. Pattern classification. 2nd Edition, Wiley- Interscience Publication. 2001.
10. Theodoridis, S. & Kourtroumbas, K. Pattern recognition. Academic Press, 1999.
11. Verma, Neelam & Shri Kant. Classification of binary patterns with parametric and nonparametric methods. Proceedings IISA 2000-2001, India, JSM, pp. 512-14.
12. Chien, Y. Interactive pattern récognition. Marcel Dekker Inc. 1978.
13. Shri Kant & Narain, Laxmi. Analysis of some stream & block ciphers: Using pattern recognition tools, NSCR' 98, 9-10 July 1998. pp. C10- C21.
14. Shri Kant. Pattern classification in context of cryptology. *In* DRDO Science Spectrum, DESIDOC, Delhi, 2002, pp. 177-98.
15. Shri Kant. What help classification techniques can provide to a cryptanalyst? *In* Proceedings of the 3rd International Conference on Pattern Recognition and Digital Technique. ISI Kolkata, Dec.1993. pp. 651-60.
16. Shri Kant. Application of pattern recognition in cryptology. *In* Proceedings of INFOSEC Bangalore, 1994, pp.104-18.
17. Shri Kant & Sharma, Veena. Discrimination among various type of encrypted bit stream. *In* Recent Development in quality reliability and information security, edited by P.K. Kapur, IMH Publisher, New Delhi, 2003, pp. 221-29.
18. Tou, J.T. & Gonzalez, R.C. Pattern recognition principles. Addison-Wesley Publishing Company, 1974.
19. Shri Kant; Sharma, Veena & Dass, B.K. On recognition of cipher bit-stream from different sources using majority voting rule. *J. Applied Mathematics, Ratio Mathematica*, 2006, **15**, 90-111.
20. Haykin, Simon. Neural networks a comprehensive foundation. Pearson Education, 4th Indian reprint, 2003.
21. Chandra, B.; Paul, P.; Saxena, P.K. & Shri Kant. Neural network for identification of crypto systems. *In* Proceedings of the 3rd Indian International Conference on Artificial Intelligence, IICAI-07, Pune, 2007, pp. 402-11.
22. Riedmiller, M. & Braun, H. A direct adaptive method for faster back propagation learning: The RPROP algorithm. *In* the IEEE International Conference on Neural Networks, 1993, **1**, pp. 586-91.
23. Vapnik, V. Statistical learning theory, Wiley, New York. 1998.
24. Burges, C.J.C. A tutorial on support vector machines for pattern recognition. *Data Mining Knowl. Discovery*, 1998, **2**(2), 955-74.
25. Vert, J.P. Introduction to support vector machines and applications to computational biology. Seminar Report, Kyoto University, Japan, 2001.
26. Dileep, A.D.; Shekhar, C. Chandra; Shri Kant & Saxena, P.K. Identification of block ciphers using support vector machines. *In* Proceedings of International Joint Conference on Neural Networks. IJCNN, (Vancouver, BC, Canada), 2006.
27. Dileep, A. D.; Sammireddy, S.; Sekhar, C. Chandra & Shri Kant. Decryption of Feistel type block ciphers using Hetro-Association model. *In* Proceedings of National Conference on Communications, NCC-08, Mumbai, Feb. 2008. pp. 74-78.
28. Sammireddy, S.; Sekhar, C. Chandra & Shri Kant. Block cipher identification using support vector classification and regression. *J. Discrete Mathematical Sci. Cryptography*, 2010, **13**(4), 305-18.
29. Shri Kant & Khan, S.S. Analyzing pseudo random bit generator through machine learning inductive algorithm. *Int. J. Intelligent Data Analy.*, 2006, **10**(6), 539-54.

Contributor

Dr Shri Kant received his PhD(Mathematics) from Banaras Hindu University. Presently working as Scientist 'G' and Coordinator, Joint Cipher Bureau, DRDO, Metcalfe House, Delhi.