# ON RECOGNITION OF CIPHER BIT STREAM FROM DIFFERENT SOURCES USING MAJORITY VOTING FUSION RULE

**SHRI KANT*, VEENA SHARMA*, B. K. DASS****

*Scientific Analysis Group
Defence R & D Organization
Metcalfe House complex
Delhi-ll0054

**Deptt. of Mathematics
Faculty of mathematical Sciences
University of Delhi
Delhi-110054

## Abstract

In the present paper, majority-voting rule has been investigated for its possible application in cryptological sciences. A novel approach is proposed to address the complex identification problem of overlapping classes. The method for representing patterns using different measurements has been discussed and the majority voting rule is used to fuse the results obtained in different measurement spaces. The proposed approach is quite natural and simple to implement in comparison with usual fusion strategies. The scheme has been implemented for three-class problem and results were tabulated and presented graphically.

**Keywords**

Decision fusion, Representation space, Pattern space, Expert classifiers, Majority logic, Stream ciphers and Cryptology.

---

**Address for correspondence:** Scientific Analysis Group, Defence R & D Organization, Metcalfe House complex, Delhi-ll0054
Tel No.: (011) 23813862
Email: shrikant@scientist.com, shrikant.ojha@gmail.com

## 1. Introduction

Identification of cipher bit streams generated from different sources is the primary step for a cryptanalyst. It requires cipher bit stream to be represented in the form of a pattern vector. In the measurement space, the analyst can take various measurements for patterns. Based on specific perception and scale, patterns are represented as points in some multidimensional feature space. The feature space is partitioned using the discriminant functions made on the basis of patterns of known classes, referred as training/learning patterns. The performance of the discriminant function is measured by categorization of independent patterns, known as test patterns, to their own partitions. A higher percentage of correct classification of the patterns in the test set indicates a better discriminator.

The fundamental goal of an analyst is to arrive at the highest probable correct classification of a given set of patterns. This objective leads to the design and development of different type of classifiers to solve a particular pattern recognition problem. Here, the accuracy in classification attained by different classifiers may be different. Also, the set of patterns correctly classified by one classifier may differ with the set of patterns correctly classified by another classifier. Thus, instead of searching for the best among the set of classifiers, it is found better to combine the decisions of individual classifiers. By applying a combination strategy to the set of classifiers such that the participating classifiers work complementary to each other, we are likely to get a classification rate better than that of a single best classifier.

Various combination strategies or decision fusion techniques have been proposed and studied by many researchers. Lam and Suen ([1]:1997), Kittler, et. al. ([2]:1998), Alkoot, et. al. ([3]:1999), Kuncheva, et. al. ([4]:2001), Chen and Cheng ([5]:2001) and Alexandre, et. al. ([6]:2001) etc. made a detailed study of different aspects of these combination strategies.

We first give a brief description of these fusion schemes. Let X be a pattern which is to be assigned to one of m possible classes $w_1, w_2, \ldots, w_m$ with the help of anyone of M individual classifiers. Each classifier approximates the *a posteriori* probability $P(w_i/X)$, i= 1, 2, … , m, that is the probability that pattern X belongs to class $w_i$, given that X was observed. A classifier assigns X to class $w_k$ if

$$P(w_k / X) = \max_{i-1,\ldots,m} P(w_i / X) \qquad ---(1)$$

For convenience, let we denote the *a posteriori* probabilities computed by classifier $C_j$ by $P_j(w_i/X)$, where j = 1, 2, …, M and i = 1, 2, …, m. It is assumed that these estimates of *a posteriori* probabilities given by individual classifiers are independent and identically distributed according to some pre-assumed distribution function.

Here, aim is to get improved estimates $P(w_i/X)$ by applying some combination rule '*f*' to the individual estimates $P_j(w_i/X)$ given by each of the M classifiers i.e.

$$P(w_i / X) = f\left(P_1(w_i / X), \ldots, P_j(w_i / X), \ldots, P_M(w_i / X)\right), i = 1, 2, \ldots, \mathrm{m} \quad ---(2)$$

Pattern X is finally allocated to class $w_k$ according to the rule (1). Thus the rule for decision fusion becomes

$$X \in w_k \quad if \quad P(w_k / X) = \max_{i=1}^{m} \left\{ f\left(P_1(w_i / X), \ldots, P_M(w_i / X)\right) \right\}$$

Some prevalent decision fusion rules are the average rule, geometric mean rule, maximum rule, minimum rule, median rule, and majority vote rule. The theoretical and experimental comparative studies about the performance of decision fusion approaches have been carried out by Kittler, et. al. ([2]:1998), Alkoot and Kittler ([3]:1999), Chen & Cheng ([5]:2001) and Kuncheva ([7]:2002) etc., using different data sets. Sensitivity to estimation errors of these schemes under different

assumptions and different approximations has been analyzed, Kittler, et. al. ([2]:1998), Alkoot and Kittler ([3]:1999). It has been found that relative performance of various combination schemes changes under different conditions. The main emphasis has been given to comparison of the two basic schemes i.e. sum rule and product rule, Kittler, et. al. ([2]:1998), Alkoot and Kittler ([3]:1999), Alexandre, et. al. ([6]:2001). The sum rule is found easy to implement and less sensitive to errors than product rule, in most of the scenarios, Kittler, et. al. ([2]:1998). The product rule and strategies devised from it perform better when all the experts produce small errors. Further, the number of classifiers employed in fusion and number of classes in the problem also has an effect on the relative performance of different experts, Alkoot and Kittler ([3]:1999).

In general as stated earlier, these fusion rules, with an exception of majority voting rule, use the probabilities obtained by different classifiers to take the final fused decision about the class-memberships of the patterns. These probabilities given by different classifiers are called soft decisions. On the other hand, majority-voting rule works on hard decisions. That is, in majority voting rule, different classifiers first give their respective decisions about the class-memberships called the hard decisions, and then the decision taken by maximum number of classifiers is taken as the final decision. Instead of handling the probabilities, it simply works on the decisions given by different classifiers and therefore, is easiest to implement, Lee and Srihari ([8]:1993) and Lam and Suen ([1]:1997). And yet, experiments show that majority-voting rule is just as effective as other combination schemes, which are more complex in nature. Also, majority-voting scheme is found to be one of the schemes, which are relatively stable.

Keeping all these facts into mind, we have chosen majority-voting scheme for experimentation to support our approach of fusion, which is slightly different from the usual approach. Let us first formulate majority-voting scheme mathematically.

In majority voting rule, the individual *a posteriori* probabilities $P_j(w_i/X)$ are used to produce hard decisions $\delta_{ij}$ where

$$\delta_{ij} = \begin{cases} 1 & \text{if} \quad P_j(w_i / X) = \max_{k=1}^{m} P_j(w_k / X) \\ 0 & \text{otherwise} \end{cases}$$

Then we assign the pattern X to class $w_k$ if

$$\left\{ \sum_{j=1}^{M} \delta_{kj} \right\} = \max_{i=1}^{m} \left\{ \sum_{j=1}^{M} \delta_{ij} \right\}$$

In the literature, mostly, two-class problem have been addressed with the help of decision fusion rules, although the rules can be implemented for any number of classes and any number of features representing a pattern. However, when the number of classes increases, the computational complexity also increases and the final decision may be costly for overlapping classes. We address this difficulty by proposing in **Section 2**, a simple and easy to implement approach, working on the basis of consensus of decisions taken in different representation spaces. **Section 3** presents the problem definition and a description about various representation spaces. **Section 4** contains the algorithm and **Section 5** contains details of experimentation and results. Finally, in **Section 6** we present our observations and conclusions.

## 2. Proposed Approach For Classification

Before discussing our approach, let us put the usual fusion approach in a form, which can be compared with proposed one. Let, there are m classes and M classifiers. As discussed before, the *a posteriori* probabilities $p_{ij}$, where i = 1, 2, …, M and j = 1, 2, …, m, are computed for a given pattern X to be classified in one of the pre-specified class.

| Pattern X | Classes | | | |
|---|---|---|---|---|
| | $w_1$ | $w_2$ | … | $w_m$ |
| **Classifier $C_1$** | $p_{11}$ | $p_{12}$ | … | $p_{1m}$ |
| **Classifier $C_2$** | $p_{21}$ | $p_{22}$ | … | $p_{2m}$ |
| : | : | : | | : |
| : | : | : | … | : |
| **Classifier $C_M$** | $p_{M1}$ | $p_{M2}$ | … | $p_{Mm}$ |

Table 2.1

The pattern X gets its class membership in class $w_i$ if a predefined function *f* as described in Section 1, gives optimum value for class $w_i$ i.e.

$$f(p_{1i}, p_{2i}, - - -, p_{Mi}) > f(p_{1j}, p_{2j}, - - -, p_{Mj}), \quad \forall\, j \neq i$$

Now, instead of considering **different types of classifiers**, we propose to consider **different representations** of same set of patterns and allow a **single classifier** to take decision about class memberships. Going this way, in spite of having only one classifier, one can have different probable decisions and can apply any of the traditional fusion schemes. Further, if one have only two or very few classifiers available, then there will be more chances of having a tie instead of having a decision due to lack of majority of a single decision, specially when we are going to deal a multi-class problem. In that case, our approach presents a way to use fusion to have more authenticated decisions by considering many representations of set of patterns, according to the underlying problem.

As stated before, we have chosen majority voting rule for fusion i.e. we accept the decision obtained in majority of the representation/feature spaces using a single classifier. Let we have 'r' representation spaces to observe a pattern X in 'r' different ways. With the help of a classifier C, we wish to classify X in one of the pre-specified m classes. Let $p_{ij}$, i = 1, 2, …, r and j = 1, 2, …, m be the probability for X of membership in $j^{th}$ class, while the $i^{th}$ representation is used to present the pattern. First we convert these soft decisions into hard decisions $\Delta_{ij,}$ by allocating one class $w_j$ to the pattern X in $i^{th}$ representation space i.e.

$$\Delta_{ij} = \begin{cases} 1 & \text{if} \quad p_{ij} = \max_{k=1}^{m} p_{ik} \\ 0 & \text{otherwise} \end{cases}$$

| Classifier C | Classes | | | |
|---|---|---|---|---|
| | **$w_1$** | **$w_2$** | --- | **$w_m$** |
| **Representation $X_1$** | $p_{11}$ | $p_{12}$ | --- | $p_{1m}$ |
| **Representation $X_2$** | $p_{21}$ | $p_{22}$ | --- | $p_{2m}$ |
| Table 2.2(a) | | | --- | |
| **Representation $X_r$** | $p_{r1}$ | $p_{r2}$ | --- | $p_{rm}$ |

**Soft Decisions:** Table2.2(a)

| Classifier C | Classes | | | |
|---|---|---|---|---|
| | **$w_1$** | **$w_2$** | --- | **$w_m$** |
| **Representation $X_1$** | $\Delta_{11}$ | $\Delta_{12}$ | --- | $\Delta_{1m}$ |
| **Representation $X_2$** | $\Delta_{21}$ | $\Delta_{22}$ | --- | $\Delta_{2m}$ |
| | | | --- | |
| **Representation $X_r$** | $\Delta_{r1}$ | $\Delta_{r2}$ | --- | $\Delta_{rm}$ |

**Hard Decision:** Table2.2(b)

From the table 2.2(b), it is clear that a pattern will get its class membership in class $w_k$ if

$$\left\{ \sum_{i=1}^{r} \Delta_{ik} \right\} = \max_{j=1}^{m} \left\{ \sum_{i=1}^{r} \Delta_{ij} \right\}$$

## 3. Problem Definition and Feature Computation

In the present day communication scenario, any type of information viz. visual scenes, voice and text, is stored and communicated digitally. The authorized recipient at the other end recovers the same with precise accuracy and correctness. The adversary may intercept, record and retrieve all the plain transmission with some trial and error, using available means and technology. But, he will not be able to make any sense of it if the information is transmitted after encipherment by applying some cryptographic techniques. To experiment with the said problem, enciphered bit streams of scenes, voice and text have been generated from three independent stream ciphers respectively. The stream ciphers used are clock-controlled shift registers, Geffe generator and cascade of linear shift registers with nonlinear combiner. The details are described in Geffe ([9]:1973), Rueppel ([10]:1986), Schneier ([11]:1996), Kumar ([12]:1997) and Menezes et. al. ([13]:1997).

We consider each fixed length sample (now onwards referred as a message) of enciphered bit stream as a pattern. These patterns require their representation in pattern space as multidimensional feature vectors so that these can become suitable for further analysis. The process of feature extraction from each message to form a suitable mathematical pattern is like an art and this is improved by experimentation and practice. Next, we will describe the procedure followed by us to extract significant feature vectors from these bit streams.

### 3.1 Mathematical Representation

Let us denote the samples of enciphered binary streams by $M^l_k$, where $l$ = 1, 2, 3 and $k$ = 1, 2, …, N. In this representation, $l$ =1 stands for encrypted scene, $l$ =2 stands for encrypted voice and $l$ =3 stands for encrypted text. The number of messages taken from each respective source is ‘N’. All messages are assumed to be of a sufficiently long length of ‘c’ bits, where $1000 \leq c \leq 5000$ bits usually. From each message $M^l_k$, binary pattern word (i.e. small blocks of bits) of a suitable fixed length ‘b’ are read,

where b = 5 or 7 bits etc. Now, these binary words can be read from a message in two (overlapping and non-overlapping) ways. In overlapped reading, we proceed bit by bit i.e. first pattern word starts from the first bit of the message and second pattern word starts from the second bit of the message and so on. And in non-overlapped reading, we move block by block i.e. we divide the whole message into blocks of given pattern word length and then these blocks are taken as pattern words.

One can take a pattern word of any length depending upon the prior knowledge of assignable character for a fixed group of bits. For a binary pattern word of length 'b', we have possibility of $2^b$ different words. If we do a certain computation on given message, for each of these $2^b$ possible words, then we will have $2^b$ computed quantities. These $2^b$ quantities or measures together will constitute a $2^b$-dimensional feature vector. So, by varying pattern word length 'b', we will get feature vectors of different dimensions from a particular message. For example, for b=5 and b=7, 32-dimensional and 128-dimensional feature vector will be obtained respectively. In each case, we get different feature space with different components and different dimensions. Following this method, we can have different representations of a particular raw pattern.

In a message $M^l_k$, number of total occurrences of pattern words,'t' is given by

$$t = \begin{cases} c-(b-1) = t_c & (\text{Overlapping case}) \\ \left[\frac{c}{b}\right] = t_d & (\text{Non-overlapping case}) \end{cases} \quad ---(3)$$

In the subsequent sub-sections, we present further, the two different types of computations done to compute the feature vectors. In these subsections, we refer '$i^{th}$ pattern word' for binary equivalent of decimal number 'i', where $0 \le i \le 2^b - 1$. For example, if b = 5 then dimension of the vector =

$2^5$ =32 and the indices of the vector will vary from 0 to 31. It can be better understood with the help of table given below.

| Binary Word | Equivalent Decimal | Feature component |
|---|---|---|
| 00000 | 0 | F[0] |
| 00001 | 1 | F[1] |
| \| | \| | \| |
| 00111 | 7 | F[7] |
| 01000 | 8 | F[8] |
| \| | \| | \| |
| 11111 | 31 | F[31] |

### 3.1.1 Percentage Frequency Vector (PFV):

First, we compute the frequency vector F. The $i^{th}$ component of the vector F, '$f_i$' is the frequency of $i^{th}$ pattern word in a particular message, where $0 \leq i \leq 2^b - 1$. So, $i^{th}$ component of the percentage frequency vector P, '$p_i$' is the percentage of the $i^{th}$ component of the frequency vector F. Each component '$p_i$' where $0 \leq i \leq 2^b - 1$, can be computed as

$$p_i = \begin{cases} \dfrac{f_i \times 100}{t_c} = p_i^c & (\text{Overlapping case}) \\ \\ \dfrac{f_i \times 100}{t_d} = p_i^d & (\text{Non - overlapping case}) \end{cases} \quad ---(4)$$

### 3.1.2 Average Distance Vector (ADV):

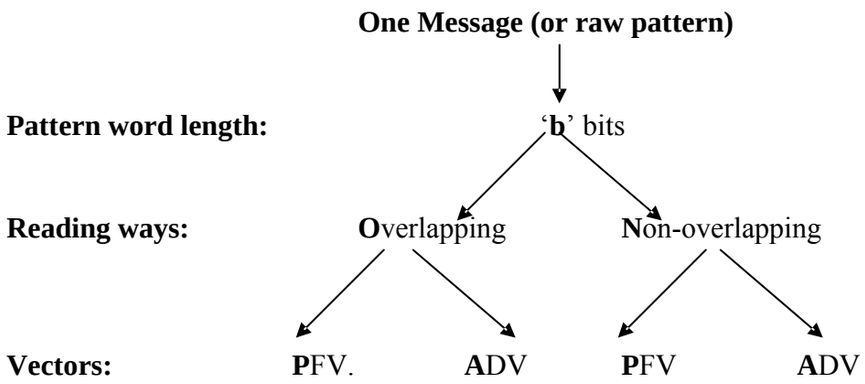
In any message, any pattern word can occur more than once. Based on these different occurrences of the same pattern word and distances between them, we compute average distance vector. Each occurrence in

the message is marked by the first bit of a pattern word. Let $P_{i,j}$ be the position of the first bit of $i^{th}$ pattern word, in $j^{th}$ occurrence in the message then the $i^{th}$ component of the average distance vector **a** is defined as

$$a_i = \frac{\sum_{j=2}^{f_i} \left( P_{i,j} - P_{i,j-1} \right)}{f_i} \qquad ---(5)$$

As defined above, $f_i$ is the number of times the $i^{th}$ pattern word occur in the message sequence, where $0 \le i \le 2^b - 1$.

Now, from each given message bit-stream for a fixed pattern word length, four types of feature vectors can be generated with the help of (4) and (5) depending upon different choices, as shown below.




In further discussion, we will use the following notations described in table 3 to refer the four possible cases. The notations have been taken as first letter of each words near the arrow in small letters.

| Notation | Pattern Word Length: b= 5 or7 | Way of Reading | Type of Vector |
|---|---|---|---|
| **bop** | '**b**' bits | **o**verlapping | **p**ercentage frequency |
| **boa** | ,, | ,, | **A**verage distance |
| **bnp** | ,, | **n**on-overlapping | **p**ercentage frequency |
| **bna** | ,, | ,, | **A**verage distance |

Table 3

## 4. Algorithm:

Suppose there are m classes $w_1$, $w_2$, ---, $w_m$ and N is the total number of raw patterns taken from each class. Thus, in total we have mN patterns. Let we denote the number of patterns to be taken for learning from each class by L. The remaining (N-L) patterns will be used for testing. Let we present all the patterns in 'r' different representations taking different combinations of choices i.e. varying the pattern word length, way of reading and type of vector. The dimension of each pattern in any representation will depend upon the pattern word length chosen for that representation. Let we denote the dimension in the $p^{th}$ representation by $n_{p,}$ where p = 1, 2, ---, r.

**Step 1:** Make a set of raw patterns (or message bit streams), keeping the patterns of all classes together. From this set, further compute 'r' sets by converting these patterns into vectors in 'r' different representations.

**Step 2:** Select one of the classification technique such as minimum distance classifier, Bayes classifier or perceptron algorithm etc. as discussed in Tou and Gonzalez ([14]:1974), Bow ([15]:1984), Kant and Sharma ([16]:2000) etc.

**Step 3:** Pass each representation of patterns to the classifier one by one i.e. for p = 1, 2, ---, r, apply classification algorithm to $p^{th}$

representation which is a set containing $n_p$-dimensional vectors. Store class allotted to each pattern in each representation.

**Step 4:** Set j = 1.

**Step 5:**For $j^{th}$ pattern, initialize count [ i ] = 0, where i = 1,2,---, m.

**Step 6:** Set p = 1.

**Step 7:** If $j^{th}$ pattern in $p^{th}$ representation goes to class $w_k$, increment the count [ k ] by 1.

**Step 8:** Repeat Step 7 for p = 2, ---, r.

**Step 9:** Finally, assign $j^{th}$ pattern to class $w_k$ if

$$\text{count [ k ]} = \max_{i=1}^{m} \{ \text{count [ i ]} \}$$

If there are more than one class such that the quantity count [k] of these classes are equal to the maximum value computed in the equation, then there arise uncertainty about the final class-membership of the pattern under consideration. In that case, the pattern is kept into the category of rejection.

**Step 10:** Repeat Step 5 to Step 9 for j = 2, ---, mN.

**5. Experimentation and Results:**

As discussed earlier, we have experimented with the problem of identification among the encrypted bit streams of scenes, speech and the text respectively. To deal with this three-class problem, we have first computed different suitable representations from these bit streams. Each representation is a set of vectors computed from the bit streams. Various techniques have been applied to classify the patterns for each

representation. Here, we are showing the classification results by two classifiers namely, maximum likelihood classifier and minimum distance classifier for each individual representation of patterns. And finally we have shown the results obtained by proposed fusion approach.

.

**Maximum likelihood classifier:** In the Tables 5.1(a) to 5.1(d), we have shown the percentage self-classification given by the maximum likelihood classifier for the four different representations of the same set of patterns. Notation used for each representation can be understood with the help of table 3. We have taken 150 patterns for learning of the classifier from each of the class. Table 5.2 shows the results obtained by fusion of classification results in individual representations. In Table 5.2, we have included the percentage of patterns, which cannot be allocated to any class due to uncertainty in deciding the final class membership.

| **% Classification** | **Representation: '5na'** | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **82.67** | 10 | 7.33 |
| **Encrypted Speech** | 8 | **84.67** | 7.33 |
| **Encrypted Text** | 8.67 | 13.33 | **78** |

Table 5.1(a)

| **% Classification** | **Representation: '5oa'** | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **82.67** | 9.33 | 10 |
| **Encrypted Speech** | 16.67 | **80** | 3.33 |
| **Encrypted Text** | 13.33 | 8.67 | **78** |

Table 5.1(b)

| **% Classification** | **Representation: '7na'** | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **98** | 0 | 2 |
| **Encrypted Speech** | 1 | **97** | 2 |
| **Encrypted Text** | 0.67 | 0 | **99.33** |

Table 5.1(c)

| % Classification | Representation: '7oa' | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **97.33** | 1.33 | 1.33 |
| **Encrypted Speech** | 1 | **97** | 2 |
| **Encrypted Text** | 0 | 4.67 | **95.33** |

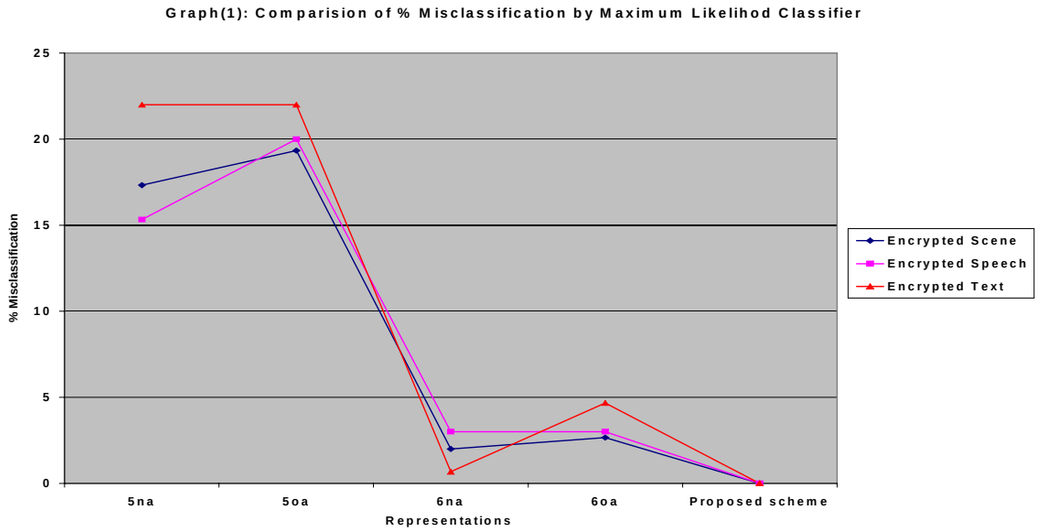
Table 5.1(d)

| % Classification | Proposed Approach | | | |
|---|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** | **Rejected** |
| **Encrypted Scene** | **96** | 0 | 0 | 4 |
| **Encrypted Speech** | 0 | **98.67** | 0 | 1.33 |
| **Encrypted Text** | 0 | 0 | **96** | 4 |

Table 5.2

In Tables 5.1(a) to 5.1(d), we observe some wrong classifications i.e. the percentage of patterns, which are misclassified to other classes to whom they do not belong actually. But in Table 5.2, we can see that there are no wrong classifications among classes, though we have some rejections here. This means that misclassification occurred in case of individual representations is somewhat corrected by our approach of fusion. And the patterns, which cannot be still correctly classified due to lack of consensus, are shifted to rejection category. Knowing that a misclassification is costly than a rejection, we found our classification approach to be advantageous.

This phenomenon is illustrated in the Graph(1) displayed next. In the graph, three series are plotted to show the percentage of number of misclassified patterns in each of the three classes. In each series, the classification results obtained in individual representations and by proposed fusion approach are compared. It is clear from the graph that using the proposed approach of fusion, we get a decrease to zero in percentage of misclassification in each of the class.




**Minimum Distance Classifier:** The percentage self-classification for different representation of patterns with minimum distance classifier has been summarized in Table 5.3(a) to 5.3(f). After fusing the classification results in these six representations, we get improved results as shown in Table 5.4. Here also, we observe that by using fusion there is a great decrement in number of misclassified patterns, in each of the class. The patterns, which cannot be allocated to any class due to a tie of votes, are kept in rejection category

| **% Classification** | **Representation: '5oa'** | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **34** | 31.33 | 34.67 |
| **Encrypted Speech** | 27.33 | **48** | 24.67 |
| **Encrypted Text** | 25.33 | 30.67 | **44** |

Table 5.3(a)

| % Classification | Representation: '5na' | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **55.33** | 18 | 26.67 |
| **Encrypted Speech** | 28.67 | **50** | 21.33 |
| **Encrypted Text** | 28.66 | 22 | **49.33** |

Table 5.3(b)

| % Classification | Representation: '7na' | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **52.67** | 24 | 23.33 |
| **Encrypted Speech** | 20 | **60** | 20 |
| **Encrypted Text** | 26 | 22 | **52** |

Table 5.3(c)

| % Classification | Representation: '7na' | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **66.67** | 20.67 | 12.67 |
| **Encrypted Speech** | 21.33 | **55.33** | 23.33 |
| **Encrypted Text** | 20.67 | 20 | **59.33** |

Table 5.3(d)

| % Classification | Representation: '5np' | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **51.33** | 25.33 | 23.33 |
| **Encrypted Speech** | 24.67 | **47.33** | 28 |
| **Encrypted Text** | 28 | 26.67 | **45.33** |

Table 5.3(e)

| % Classification | Representation: '7np' | | |
|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** |
| **Encrypted Scene** | **66.67** | 16.67 | 16.67 |
| **Encrypted Speech** | 16.67 | **64.67** | 18.67 |
| **Encrypted Text** | 20 | 16 | **64** |

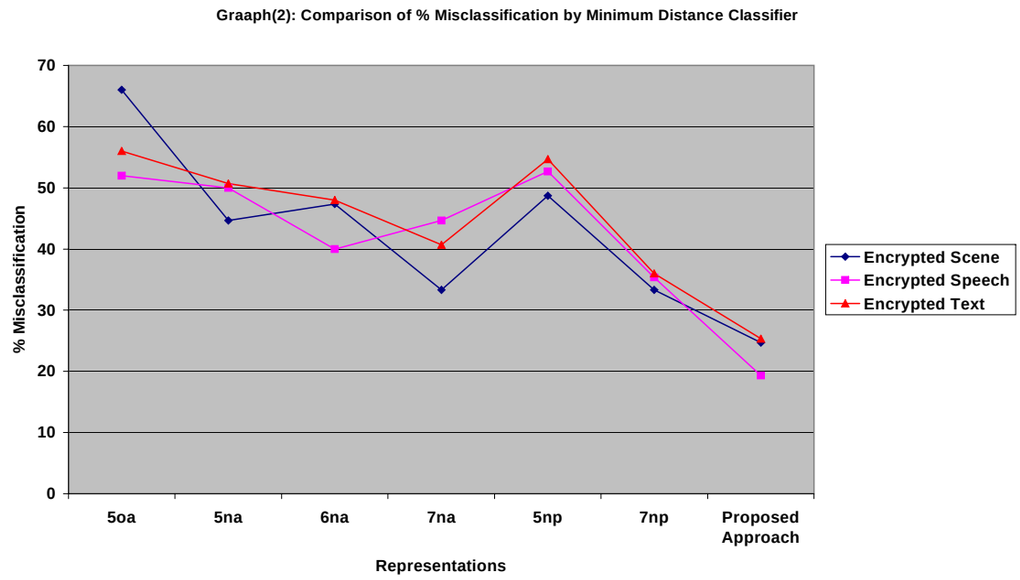
Table 5.3(f)

| % Classification | Proposed Approach | | | |
|---|---|---|---|---|
| | **Encrypted Scene** | **Encrypted Speech** | **Encrypted Text** | **Rejected** |
| **Encrypted Scene** | **62.67** | 12.67 | 12 | 12.67 |
| **Encrypted Speech** | 8 | **60** | 11.33 | 20.67 |
| **Encrypted Text** | 14.67 | 10.67 | **59.33** | 15.33 |

Table 5.4

With minimum distance classifier, we are able to get more than 55% classification consistently for each class. These results of classification may not be very high for the practical application, but this consistency is extremely useful from a cryptanalysis point of view.

The Graph(2) plotted to compare the results by minimum distance classifier, in different representations and those obtained by fusion, presents a similar trend as shown by maximum likelihood classifier. As compared to individual representations, the proposed fusion approach gives the least number of wrongly classified patterns.




Using both the classifiers, we have tested several sets of patterns from each class, and the test-classification is also found to be quite encouraging.

## 6. Observations and Conclusion:

The experimentation done for the present work has given us enough idea about handling the problem of discrimination among various random sources. The proposed idea is quite general in nature and can be applied to other kind of classification problem as well, if it is possible to compute different measurements for the same set of patterns. Also, experimentation can be done for any number of classes as described in Algorithm, instead of restricting to a three class problem. According to the nature of underlying problem and knowledge of significant features, different measurements may be computed to get different representations of

patterns. For our problem, we adhered to the most suitable representations of patterns where the classification is more transparent. Dealing with the said problem, the following observations and constraints are found to be important:

1. While assigning class membership to a pattern in each of the representation space, the classifier has three possibilities. Either the pattern will be correctly classified, wrongly classified or the classifier will remain uncertain about the class membership of the pattern. The third possibility of uncertainty of decision arises due to tie between values of discriminating function for the possible classes. This situation of neutral position of the classifier leads to no decision or rejection, i.e. classification is neither correct, nor wrong. Here, for convenience, we have considered only those representation spaces in which classifier had only two alternatives, of being correct or wrong and no rejections. Again, while taking the final decision by fusion as proposed, there may be cases of no consensus. This situation of uncertainty in deciding final class membership of a pattern leads to a rejection. We have kept these patterns in a separate category.

2. After applying proposed fusion, it has been observed that there are no wrong classifications with maximum likelihood classifier, though there are few patterns, which cannot be allocated to any class and have been kept in no decision category. Minimum distance classifier shows the similar trend with less wrong classifications by fusion as compared to those obtained in individual representations. Here also, the patterns about which the classifier is not certain are kept in rejected category. For both the classifiers, graphs have also been plotted to compare the percentage misclassification of patterns, in individual representations and after fusion by proposed approach. It is clear from the graphs by using the proposed fusion approach that we are getting reduced percentage of misclassification, which is the merit of our approach.

3. Final results, obtained by fusion by proposed approach, are better than the results obtained by using single representation spaces.

4. As we have discussed earlier, each of the representation space has dimension as $n_p$, p = 1, 2, ---, r. The general observation is that we obtain consistently better performance when the size of learning set is more than 5x $n_p$.

**Acknowledgement**

We would like to express our sincere gratitude and deep veneration to DR. P K Saxena**,** Director SAG and Dr. Laxmi Narain Sc. 'F' for giving us this opportunity to carry out the present work. We are also thankful to Ms. Neelam Verma, Sc. 'E' for her constructive suggestion made during the preparation this paper.

**References:**

[1]. Lam, L., Suen, C. Y., 1997. Application of majority voting to pattern recognition: an analysis of its behaviour and performance. IEEE transactions on Systems, Man, and Cybernetics 27(5), 553-568.

[2]. Kittler, J., Hatef, M., Duin, R., Matas, J., 1998. On combining classifiers. IEEE Trans. PAMI 20(3), 226-239.

[3]. Alkoot, F. M., Kittler, J., 1999. Experimental evaluation of expert fusion strategies. Pattern Recognition Letters 20, 1361-1369.

[4]. Kuncheva, L., Bazdek, J., Duin, R., 2001. Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition Letters 34(2), 299-314.

[5]. Chen, D., Cheng, X., 2001. An asymptotic analysis of some expert fusion methods. Pattern Recognition Letters 22, 901-904.

[6]. Alexandre, Luis A., Campilho, Aurelio C., Kamel, M., 2001. On combining classifiers using sum and product rules. Pattern Recognition Letters 22, 1283-1289.

[7]. Kuncheva, Ludmilla I., 2002. A theoretical study on six classifier fusion strategies. IEEE Trans. PAMI 24(2), 281-286.

[8]. Lee, D. S., Srihari, S. N., 1993. Handprinted digit recognition: A comparison of algorithms. In Proc. 3rd Int. Workshop Frontiers Handwriting Recognition. Buffalo, NY, pp. 153-162.

[9]. Geffe, P. R., 1973. How to protect data with ciphers that are really hard to break. Electronics, 46(1), 99-101.

[10]. Rueppel, R. A., 1986. Analysis & design of stream ciphers. Springer-Verlag.

[11]. Schneier, B., 1996. Applied cryptography, Second Edition John Wiley & Sons, Inc.

[12]. Kumar, I. J., 1997. Cryptology: System identification and key clustering. Agean Park Press, CA, USA.

[13]. Menezes, A. J., Van Oorschot, P. C., Vanstone, S. A., 1997. Handbook applied cryptography. CRC Press, Boca Raton.

[14]. Tou, J. T., Gonzalez, R. C., 1974. Pattern recognition principles. Addison-Wesley Publishing Company.

[15]. Bow, Sing-Tze, 1984. Pattern recognition: Application to large data-set problems. Marcel Dekker, Inc., New York & Basel.

[16]. Kant, S., Sharma, V., 2000. Discrimination among various type of encrypted bitstream. International Conference on Quality Reliability and Information Technology, 21-23 Dec, New Delhi.