Reduct ECOC Framework for Network Intrusion Detection System

Uma Shankar Rao Erothi, Sireesha Rodda

***Abstract:** Now a day's network security is major concern for e-government and e-commerce applications. A wide range of malicious activities are increasing with the usage of internet and network technologies. Identifying novel threats and finding modern solutions for network to prevent from these threats are important. Designing an effective intrusion detection system is significant to continuously look out the network activities to efficiently thwart malicious attacks or to identify the intruders. To tackle multi class imbalance classification problem in networks, a reduct based ECOC ensemble framework for NIDS is proposed to efficiently identify attacks in a multi class scenario. The Reduct-ECOC classifier is validated on highly imbalanced benchmark NSL-KDD intrusion datasets as well as other UCI-ML datasets. The experimental results on eight highly imbalanced datasets show that Reduct-ECOC classifier performs better than many other state-of-art multi-class classification ECOC learning methods.*

***Keywords:** Network Intrusion Detection System, Multi Class Imbalance, Rough Set Theory, ECOC Classifier.*

I.INTRODUCTION

The fast-paced improvements of internet and network applications in different fields increases the number of network users and usage of computers. It is necessary to safeguard the health of network applications in spite of ever increasing malicious activities on the internet. It emerges the development of modern network security- solutions to identify novel threats [1]. Most of the traditional approaches for intrusion detection can be implemented using anomaly and misuse/signature based detection. Misuse detection monitor's packets on network and matches them against database of known threats/signatures. Generally, this approach good at detecting known attacks, but fails to detect novel attacks. Whereas, the anomaly detection builds model on normal traffic patterns, any deviation from these patterns are committed to false alarm [2]. The popular intrusion datasets such as DARPA'98 [3], KDD_CUP'99 [4] and NSL_KDD [5] are used to evaluate intrusion detection system (IDS). The standard machine learning (ML) algorithms such as decision tree (J48/C4.5/CART), Support Vector Machine, AdaBoost, K-Nearest Neighbor, Naïve Bayes etc., are alone not applicable to solve the misuse related problems due to several specific reasons that includes skewed class distribution. The traditional classifiers, tend to predict only majority class and treats the minority class as noise, due to the number of attacks traffic when compared to normal traffic is quite less [6].

The problem of skewed class distribution in the network intrusion detection causes class imbalance problem. The traditional classifiers fails to identify the attack categories such as Dos, Probe, U2R & R2L.The results obtained from the most of the classifiers for a particular dataset would see greater accuracy for the majority class, but poor performance on detecting the minority classes [6]. Working with sheer volume of network connection data expose computational complexity, slow down the classification process, and leads to unsatisfactory results. Traditional data mining and machine learning algorithms are facing difficulty in dealing with highly imbalanced data. Many real-world applications are multi-class in nature and their class distribution in a dataset varies from one class to other. To address class imbalance and multi class imbalance problems, several approaches have been proposed such as tree based approach [7], multi class kernel based support vector machine [8], boosting [9], random forests [10], and error correcting output codes (ECOC) [11].The study of multi-class imbalance classification problem is more challenging than binary-class imbalance.

ECOC is a successful ensemble learning framework, it decomposes the multi class problem into a small set of binary classification problem [12, 13]. Several ECOC approaches such as basic ECOC, Fuzzy ECOC [28], forest-ECOC [15], Discriminant-ECOC [16], node embedding ECOC, deep learning ECOC [18] and imbalanced ECOC (imECOC) have been suggested to tackle multi class imbalance problem. However, these approaches needs sophisticated coding and decoding strategies, they fall under two categories problem independent (PI) and problem dependent (PD) methods. The variations of the above ECOC methods for classification tasks are confined to binary $\{-1,1\}$ and ternary codes {-1, 0, 1}.Therefore, the effective coding schemes for ECOC ensemble learning exploit it's idea to combine with that of deep learning methods presented in [20]. Several deep learning models have been proposed to handle various real world challenges such as image detection [21], text analysis [22], video analysis etc. The simplicity of ECOC learning models are beneficial for the large scale applications, class imbalance and multi class imbalance problems.

Revised Manuscript Received on February 05, 2020.

* Correspondence Author

Uma Shankar Rao Erothi*, Department of CSE, RAGHU Institute of Technology, Visakhapatnam, India.

E-mail: umashankar.erothi3@gmail.com

Sireesha Rodda, Department of CSE, GITAM University, Visakhapatnam, India, E-mail: sireesha.rodda@gitam.edu

Retrieval Number: B4238129219/2020©BEIESP
DOI: 10.35940/ijeat.B4238.029320
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

II.OVERVIEW OF METHODS

ECOC

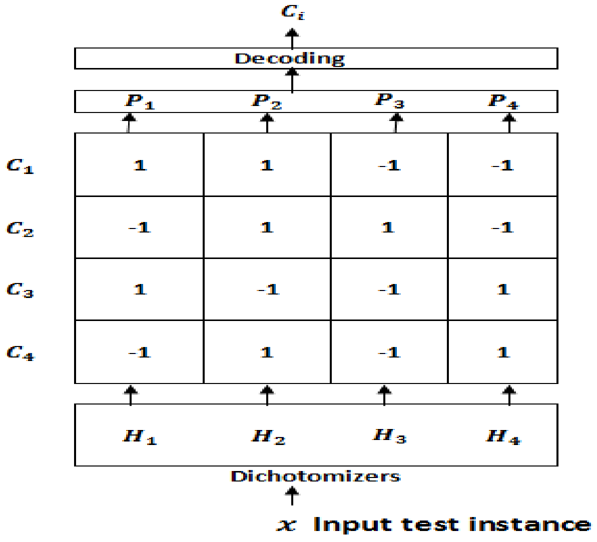
The basic idea of the ECOC ensemble learning classifier is to discriminate instances that belongs to different classes using binary or ternary coding mechanisms shown in Figure.1 and Figure.2. The basic ECOC framework has three steps-Coding, Learning, and Decoding.

In coding stage, for a given training dataset $D = \{(x_i, f(y_i)\}_{i=1}^{n}$, with x_i belongs to conditional attributes and $f(y_i)$ be longs to decision attribute with N_C number of classes $\{C_1, C_2, C_3, \dots, C_N\}$. This approach decompose given set of N_C classes into a small set of l binary problems with the unique code-words [17] in code-matrix $M \in \{+1,-1\}^{Nc*l}$ or $M \in \{+1,0,-1\}^{Nc*l}$. Then, each two-class (column) partitions (D^i) are trained by the respective dichotomizer (H_i) to obtain a new code word $P_i \in \{+1,-1\}$ or $\{+1,0,-1\}$. During decoding process, Y_i is assigned to class C_i with the nearest code-word in Matrix M using Hamming Distance (HD) or Euclidean Distance (ED) shown in equation (1) and (2). For Example, the ECOC coding design shown in figure (1) and (2) show how the classes are partitioned in the case of binary and ternary, where +1 indicates considered(positive) classes for the respective dichotomizer, -1 indicates negative classes and 0 indicate class that are not considered in the learning shown in eq.(3) and eq.(4). The base classifiers or dichotomizers $\{H_1, H_2, H_3, H_4\}$ are trained on each partition, at the decoding stage a new test instance X_i is evaluated by n dichotomizers, the t^{th} bit distance of example X_i to class C_i is defined as shown in eq.(3) to obtain a solution vector$\{P_1, P_2, P_3, P_4\}$. Then classify the new test instance by the class C_i which code-word minimizes the decoding measure.

$$HD(C_i, P_i) = \sum_{i=1}^{n} \frac{|C_i - P_i|}{2} \qquad (1)$$

$$ED(C_i, P_i) = \sqrt{\sum_{i=1}^{n} (C_i - P_i)^2} \qquad (2)$$

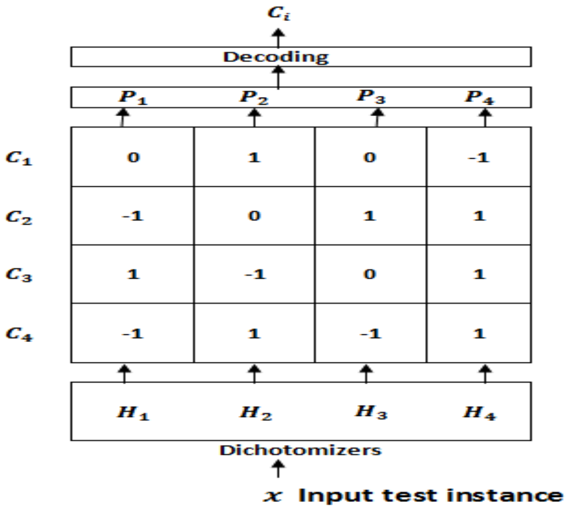
$$d_{M(i,t)} = \{P_i\}_{i=1}^{N_C} \qquad (3)$$



	P_1	P_2	P_3	P_4
C_1	1	1	-1	-1
C_2	-1	1	1	-1
C_3	1	-1	-1	1
C_4	-1	1	-1	1

Fig. 1. Binary ECOC

$$H_1(x) = \begin{cases} 1 \quad if\ x \ \in \{c_1, c_3\} \\ -1\ if\ x \ \in \{c_2, c_4\} \end{cases} \qquad (4)$$



	P_1	P_2	P_3	P_4
C_1	0	1	0	-1
C_2	-1	0	1	1
C_3	1	-1	0	1
C_4	-1	1	-1	1

Fig. 2. Ternary ECOC

$$H_1(x) = \begin{cases} 1 & if\ x \ \in \{c_3\} \\ -1 & if\ x \ \in \{c_2, c_4\} \\ 0 & Not\ considered \end{cases} \qquad (5)$$

Algorithm 1:ECOC_Algorithm(*D*,N_c)

Input: Training dataset D with N_c number of classes

Dichotomy classifier H_i on sub-partition D^i

Code Matrix : $M \in \{+1,0,-1\}^{Nc*l}$ or $M \in \{+1,-1\}^{Nc*l}$

Solution Vector : $d_{M(i,t)} = \{P_i\}_{i=1}^{n}$

Output: ECOC classifier

Training Phase

```
1  Generate $N_C * l$ distinct binary code word
   matrix $M$
2  Assigning unique code to each $C_i$ from code
   matrix $M_i$, where i={1,2,…,n}
      for i:= 1 to $N_C$ do
         $C_i \leftarrow M_i$
      end for
3  Train all the base classifiers H$_i$ to learn $N_C$
   binary functions i.e., one for each column.
      for i = 1 to $N_C$ do
         $H_i \leftarrow l(D^i, C_i)$
      end for
4  Apply each dichotomizer (H$_i$) classifiers to the
   given test example
      for i = 1 to $N_C$ do
         $H_i(x_i, r), \forall x_i \in D^i, \forall r = 1,2,\dots,N_C$
      end for
```

Testing Phase

```
5  Classify new test sample with the nearest code
   word ($C_i$) using HD or ED
   Combine all the predictions to form a new output
   vector $d_{M(i,t)}$ of length $l$.
6  evaluate measures
```

IMECOC

The work proposed in [19], tackles the problem of multi class imbalance to solve the learning difficulty in both between-class and with-in class imbalance among majority and minority classes of a dichotomy shown in below example. The positive class and negative class in a dichotomy is generally imbalanced (called between-class imbalance) because the original input space is imbalanced. The subclasses in positive and negative are differ in sizes (called within-class imbalance). The weighted decoding mechanism for each dichotomy classifiers minimizes the false alarm rate in the case of minority classes. This method outrages many other state-of-the-art multi class imbalance learning methods. The proposed Reduct-ECOC use same weighted decoding mechanism of imECOC approach shown in algorithm 2.

Example: Let us consider a dataset with N_4 number of classes $\{C_1, C_2, C_3, C_4\}$, and its distribution $\{n_1 = 10, n_2 = 50, n_3 = 100, n_4 = 1000\}$, for a given dichotomy code [+1,-1, +1,-1], between-class imbalance is considered as imbalance between the total size of $|A_+| = 110 \; and \; |A_-| = 1050$, i.e., $\frac{|A_-|}{|A_+|}$ = 9.55, with-in class imbalance is considered as imbalance between sub-classes $C_1 \; and \; C_3$, i.e., $\frac{n_3}{n_1} = 10$, and between $\{C_2, C_4\}$, i.e., $\frac{n_4}{n_2} = 20$.

Algorithm 2:imECOC_Algorithm(D,N_c)

```
Input : Training dataset D with $N_c$ number of classes
Dichotomy classifier $H_i$ on sub-partition
Code Matrix : $M \in \{+1,0,-1\}^{N_C * l}$
BWC-weighting method
Output :  imECOC classifier
Training phase
1   Generate $N_C * l$ distinct binary code word
    matrix $M$
2   Generate Dichotomy datasets($D^i$) w.r.t $C_i$ from
    code matrix $M_i$, where i={1,2,…,n}
        for i:= 1 to  $N_C$ do
            $D^i = \emptyset$
            for j:= 1 to n do
                if  $M(y_j, H_j) \neq 0$ then
                    $D^j = D^j \cup (X_j, M(y_j, H_j))$
                end if
            end for
3   Calculate BWC- weights for the class $C_i$, for the
    $H_i$ dichotomy
4   Apply $\mathrm{H}_i$ learned classifiers using weighted
    code-matrix
        for i = 1 to  $N_C$ do
          $H_i \leftarrow l(D^i, C_i)$
        end for
5   Calculate distance vector for the $D^i$ training set,
    to obtain optimal weights for a given dichotomy.
        end for
        Testing phase
6   Apply $\mathrm{H}_i$ learned classifiers to the test example
        for i = 1 to  $N_C$ do
            $H_i(x_i, r), \forall x_i \in D^i, \forall r = 1,2,\dots,N_C$
        end for
7   Classify new test sample with the nearest code
    word ($C_i$)
8   evaluate measures
```

RSS-ECOC

The work proposed in [23] is based on feature/attribute selection space in the design process of the ECOC matrix. That is, each dichotomize is trained on a different feature subset to obtain better classification accuracy. From the design process point of view, a 3D code-matrix is generated, where the third dimension is the feature space of the problem domain. To generate ECOC framework, a 2D code matrix is first generated from a previous set of matrices that maximizes the minimum distances between any pair of code words.

Algorithm 3:RSS-ECOC_Algorithm(D,N_c)

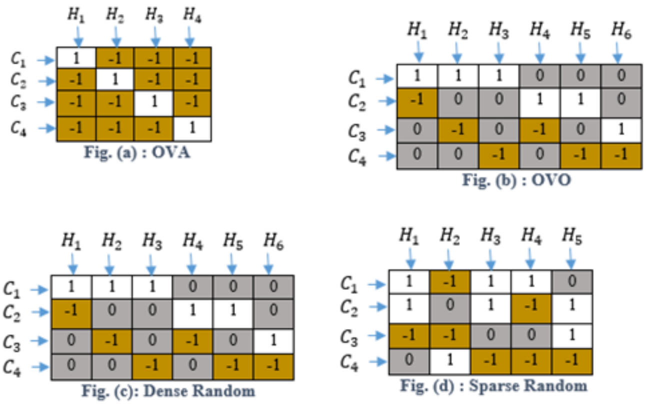
```
Input: Training set $D = \{(x_i, H(y_i)\}_{i=1}^{n}$;
$H(y_i) = \{1,2,..,\mathrm{N_C}\}$; binary classifier $l$
$D$ is the original feature space with $A_i$ attributes
   $f_i = \{A_1, A_2, \dots, A_n\}$ after feature selection
Code Matrix : $M \in \{+1,0,-1\}^{N_C * l}$   or  $M \in \{+1,-1\}^{N_C * l}$
Output :  RSS-ECOC classifier
Preprocessing phase
1   Input dataset D
2   Feature selection using QuickMultipleReduct
    algorithm to obtain new subspaces $f_i$
Training Phase
3   Generate $N_C * l$ distinct binary longer code word
    matrix $M$
4   Generate Dichotomy datasets($D^i$) w.r.t $C_i$ from
    code matrix $M_i$, where  i={1,2,…,n}
5   Apply each of the $\mathrm{H}_i$ learned dichotomizers using
    CART & MLP as a base classifier
        for i = 1 to  $N_C$ do
          $H_i \leftarrow l(D^i, C_i)$
        end for

Testing phase
6   Apply $\mathrm{H}_i$ learned classifiers to the test example
        for i = 1 to  $N_C$ do
            $H_i(x_i, r), \forall x_i \in D^i, \forall r = 1,2,\dots,N_C$
        end for

7   Classify new test sample with the nearest code
    word ($C_i$) using Exponential Loss-Weighted
    (ELW) decoding
8       evaluate measures
```

CODING AND DECODING STRATEGIES

A successful ECOC ensemble framework need a proper coding and decoding implementation to improve generalization ability, to reduce the bias and variance produced by the learning algorithms. The code-words for N_C class should satisfy the rows and columns are well separated in terms of distance metrics i.e., HD and ED.

The created code words cannot guarantee that are always discriminative for N_C class classification task.

The estimated code length requires $15 * log_2\,(N_C)$ and $10 * log_2(N_C)$ for Sparse and Dense random approaches shown in Fig. 3(c) and 3(d).To increase error correcting capability and diversity among base classifiers have been reported in [14]. The experimental results obtained on 10-benchmark datasets with One Versus One (OVO) method uses $N_C(N_C-1)/2$ base classifiers and One Versus All (OVA) with N_C-1 dichotomies [24] shown in Fig. 3(a) and 3(b).



	H_1	H_2	H_3	H_4
C_1	1	-1	-1	-1
C_2	-1	1	-1	-1
C_3	-1	-1	1	-1
C_4	-1	-1	-1	1

Fig. (a) : OVA

	H_1	H_2	H_3	H_4	H_5	H_6
C_1	1	1	1	0	0	0
C_2	-1	0	0	1	1	0
C_3	0	-1	0	-1	0	1
C_4	0	0	-1	0	-1	-1

Fig. (b) : OVO

	H_1	H_2	H_3	H_4	H_5	H_6
C_1	1	1	1	0	0	0
C_2	-1	0	0	1	1	0
C_3	0	-1	0	-1	0	1
C_4	0	0	-1	0	-1	-1

Fig. (c): Dense Random

	H_1	H_2	H_3	H_4	H_5
C_1	1	-1	1	1	0
C_2	1	0	1	-1	1
C_3	-1	-1	0	0	1
C_4	0	1	-1	-1	-1

Fig. (d) : Sparse Random

Fig 3: Variations of ECOC Code Matrix

The exhaustive code length for 10 and 5-class with the row separation of HD 8 can be found in [11]. The loss based [14] decoding strategy between H_i trained dichomizer, response vector $P_i=\{P_1,P_2,\ldots,P_n\}$ and the base code words C_i is formulated as equation (6), Where $L(.)$ denotes the loss function.

$$LB(P_i,C_i)=\frac{1}{2}\sum_{i=1}^{n}L(P_i*C_i) \qquad (6)$$

The work proposed in [17] improved the performance of the ECOC framework using Attenuated Euclidean decoding strategy to deal with the problem of zero symbols shown in Equation 7.

$$d_j=\sqrt{\sum_{i=1}^{n}\left|C_i^j\right|(P_i-C_j^i)^2} \qquad (7)$$

Where d_j indicates distance from the row j , n indicates number of dichotomies, P_i is the response vector of the base classifiers over test example, and C_i^j is the binary values of code matrix shown in figure 1 and 2, at the i^{th} row and j^{th} column respectively.

ROUGHSET THEORY

Rough set theory (RST) [25] is a mathematical approach that supports approximation in decision making. A decision system' (S) is defined as set of conditional- attribute (C) and decision-attribute (D). Usually, RST can be used to obtain minimal feature subset by comparing equivalence relation generated by a set of feature called "reduct". The reducts generated from RST has similar capability to classify objects as in the complete training set of C. The proposed framework utilized quick reduct algorithm (QRA) [26], QRA is used to generate multiple possible reducts for a given training dataset by constructing discernibility function on the input dataset. The detailed working of QRA is presented in Algorithm 4. Line 1 starts with an empty reduct-set (∅) and line 5 through line 7, selects the features having greatest rough set dependency metric. This process repeated until features produces its maximum possible values for the given dataset.

Algorithm 4:QuickReduct_Algorithm(C,D)

```
Input   : Decision system : S
Output : reduct set : Ŕ
1   Ŕ ← {∅}
2   while  $\gamma_{Ŕ}$ (D) ≠ $\gamma_C$ (D)
3       S ← Ŕ
4      ∀x ∈ ( C − Ŕ)
5       if $\gamma_{Ŕ\,\cup\,\{x\}}$ (D) > $\gamma_S$ (D)
6   S ← Ŕ ∪ {x}
7    Ŕ ← S
8       end while
9       return (Ŕ)
```

PROPOSED METHOD

The focus of the proposed Reduct-ECOC ensemble classifier is to provide high detection rate with low FAR. The framework presented in Figure. (4) Operates in two phases: - classification and validation phase.

In classification phase, the original input space is partitioned into a set of minimal subspaces using QRA, to generate maximum possible reducts shown in algorithm 4. Once the reduct subspaces are obtained, Reduct-ECOC with base learner C4.5 is trained on each reduct shown in algorithm 5.

In coding stage (line 2), we considered dense random binary coding design shown in Figure.1, each dichotomizer is trained to distinguish one-class from the rest of the class. Let $\{C_1,C_2,\ldots,C_n\}$ be n binary strings of length l for each of N_C classes, the HD between every pair of codeword's C_i is as large as possible (the number of bit positions differ in codeword's). We will call each string C_i be the code word for final predicted class belongs to N_C , where $i=\{1,2,..,n\}$. This method uses codeword's as rows of a matrix M, where $M\in\{-1,+1\}^{N_C*l}$, N_C indicates number of classes and l is the length of the codeword.

Reduct-ECOC (line 3) is an ensemble classifier which combine many binary classifiers to solve the multi-class classification problem. The task of any learning algorithm from examples is to find an appropriate definition for an unknown function $f(y_i)$ for a given training examples of the form $(x_i,f(y_i))$ that can solve two class problems. In learning, the binary classifiers are trained on each sub partitions of the classes in the columns of code matrix M as shown in algorithm 2(line 2). During learning each of the H_i dichotomizer is learned by re-coding the examples to be $\{(x_i,f(y_i)\}_{i=1}^{n}$ and applying a binary classifier learning algorithm to learn $H^*{}_i$. The result of this is a vector of N_c hypothesis$\{H^*{}_1,H^*{}_2,\ldots,H^*{}_l\}$.

In decoding step shown in line 5, each dichotomy classifier predict a value for a given test sample resulting in the code word of length l. Then the test instance is assigned to the closest class in the code-matrix M using hamming decoding shown in line 6.

In Validation phase, to classify new test sample X_i of each learned function $f(y_i)$ to X_i to return a response vector from binary classifiers $P^* = \{H^*{}_1(x^1), H^*{}_2(x^2), \dots, H^*{}_k(x^l)\}$. Then find code word P_i closest to this $C_i$vector using HD, shown in line4 through 6.

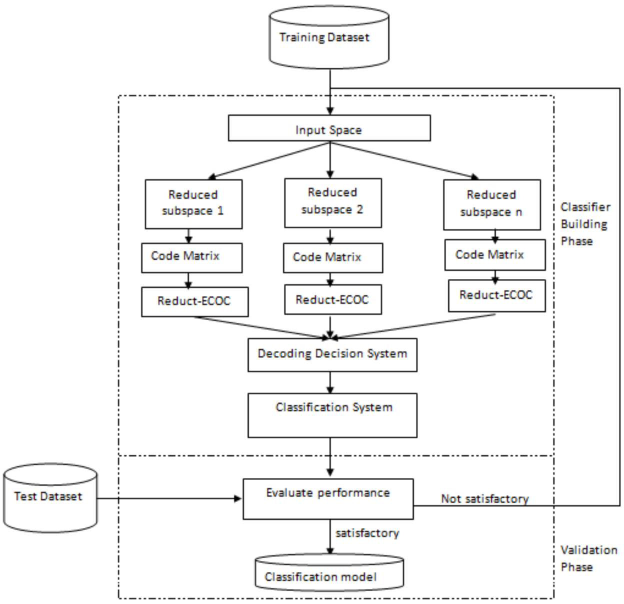


Fig. 4. Reduct ECOC Framework for NIDS

Algorithm 5 : Reduct- ECOC algorithm

```
Input :Training set $D = \{(x_i, f(y_i)\}_{i=1}^{n}$ ;
$f(y_i) = \{1,2,..,N_C\}$; binary classifier $l$
#Pre-Processing phase
1   Training set $D = \{(x_i, H(y_i)\}_{i=1}^{n}$
2   Reduced sub-spaces using algorithm 4 to obtain
    un-correlated features

#Classification Phase

1   Generate $N_C * l$ distinct binary code word
    matrix$M$, where $M \in \{-1,+1\}^{\mathrm{Nc}*l}$ in binary
    case
2   Each class assigned one row of matrix $M$
        for i = 1 to $N_C$
            $C_i \leftarrow M_i$
        end for
3   Train dichotomities with base classifier $\mathrm{H}_i$ to
    learn $N_C$ binary functions i.e., one for each
    column, using C 4.5
        for i = 1 to l
            $\mathrm{H}_i \leftarrow l(D^i, C^i)$
        end for

#Validation phase

4   Apply $\mathrm{H}_i$ learned classifiers to the test example
        for i = 1 to l
        $C(x_i, r), \forall x_i \in D, \forall r = 1,2,\dots,N_C$
5   Combine all the predictions to form a vector $(\mathrm{H}_i)$
    of length $l$.
6   Classify new test sample with the nearest code
    word $(C_i)$ using HD
            $\mathrm{H}^*(x) = argmin_r C(x_i, r)$
7   evaluate measures
8   If measures not found satisfactory
9   repeat the process until results are satisfactory
```

III.DATASET DESCRIPTION

The performance of Reduct-ECOC ensemble framework is evaluated on benchmark NSL-KDD dataset. Since 1999, many researchers used KDD_CUP99 intrusion dataset for anomaly based detection approach. It has millions of records collected from DARPA98 intrusion dataset, which consists of 4-GB tcp network dump. NSL is another improved version of KDD'99 dataset, it eliminate redundant connections from the training and test dataset. NSLKDD dataset contains 41 conditional attributes and one decision attribute, each record is labeled as either normal or an abnormal (attack) connections. The number of records in NSL_KDD for different types of attack categories is provided in Figure. (5), and attacks are categorized into four types: Probe, DOS, R2L and U2R as shown in Figure. (5), the details of other imbalanced benchmark datasets from University-of-California-Irvine-Machine-Learning (UCI-ML) Repository [27] are presented in Table 1.

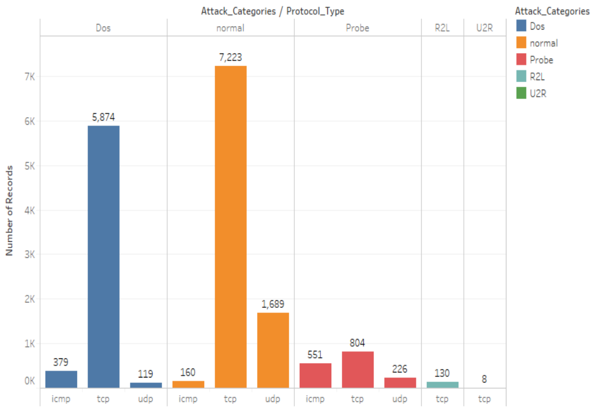


Fig. 5: Detailed Summary of NSL_KDD class distribution

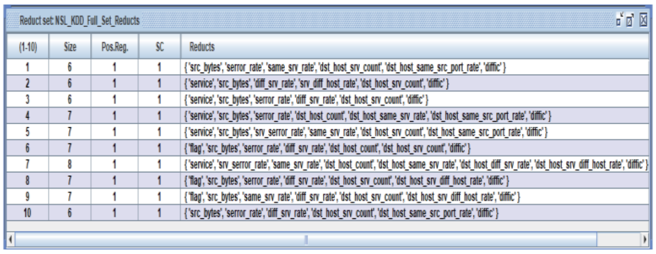
Table 1: Summary of UCI-ML datasets

Dataset	#TRAIN	#Attribute	#Class
Iris	150	4	3
Car	1728	6	4
Balance Scale	625	4	3
Glass	214	10	7
Wine	178	13	3
Zoo	101	17	7
NSL_KDD	17163	42	22

IV.EXPERIMENTAL RESULTS AND ANALYSIS

All the experiments were conducted on Intel. ® Core ™ i3-5005 CPU @ 2.00GHz Personal Computer with 8GB of RAM running on 64 bit OS. The implementation is done in Python and Java programming languages. The experiments were performed on all the benchmark UCI-ML datasets and NSL_KDD intrusion dataset.

The Reduct-ECOC classification framework presented in this paper are implemented with dense random coding strategie, the code length in code matrix is $(2^{N_c-1} - 1)$. Dichotomizers in Reduct-ECOC design are trained using C4.5 as base learner. The decoding strategy for all the ECOC schemes is chosen based on minimum HD. To reduce the computational complexity of the experiments in the Reduct-ECOC design, we considered maximum 10 reducts for NSL_KDD dataset when QRA generated more than 10 reducts shown in fig.6.The performance of Reduct-ECOC is compared against Basic-ECOC, imECOC and R-ECOC methods for handling imbalanced multi class data. All ECOC schemes presented in this paper use same code matrix, and the consistency of these methods are evaluated by considering all the measures such as Accuracy, Precision, DetectionRate (DR), False Alarm Rate (FAR), F_Score ,G_mean, and Area under Curve (AUC).


Reduct set: NSL_KDD_Full_Set_Reducts

(1-10)	Size	Pos.Reg.	SC	Reducts
1	6	1	1	{'src_bytes', 'serror_rate', 'same_srv_rate', 'dst_host_srv_count', 'dst_host_same_src_port_rate', 'diffic'}
2	6	1	1	{'service', 'src_bytes', 'diff_srv_rate', 'srv_diff_host_rate', 'dst_host_srv_count', 'diffic'}
3	6	1	1	{'service', 'src_bytes', 'serror_rate', 'diff_srv_rate', 'dst_host_srv_count', 'diffic'}
4	7	1	1	{'service', 'src_bytes', 'serror_rate', 'dst_host_count', 'dst_host_same_srv_rate', 'dst_host_same_src_port_rate', 'diffic'}
5	7	1	1	{'service', 'src_bytes', 'srv_serror_rate', 'same_srv_rate', 'dst_host_srv_count', 'dst_host_same_src_port_rate', 'diffic'}
6	7	1	1	{'flag', 'src_bytes', 'serror_rate', 'diff_srv_rate', 'dst_host_count', 'dst_host_srv_count', 'diffic'}
7	8	1	1	{'service', 'srv_serror_rate', 'same_srv_rate', 'dst_host_count', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_host_srv_diff_host_rate', 'diffic'}
8	7	1	1	{'flag', 'src_bytes', 'serror_rate', 'diff_srv_rate', 'dst_host_srv_count', 'dst_host_srv_diff_host_rate', 'diffic'}
9	7	1	1	{'flag', 'src_bytes', 'same_srv_rate', 'diff_srv_rate', 'dst_host_srv_count', 'dst_host_srv_diff_host_rate', 'diffic'}
10	6	1	1	{'src_bytes', 'serror_rate', 'diff_srv_rate', 'dst_host_srv_count', 'dst_host_same_src_port_rate', 'diffic'}

Fig.6: NSL_KDD Full set reducts

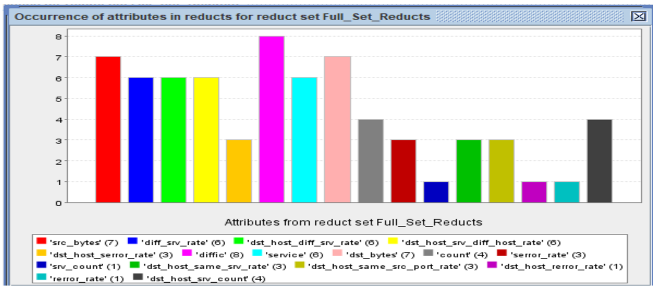


Fig.7: Occurrence of NSL_KDD attributes in reducts

Handling irrelevant and redundant features in high dimensional dataset causes a long term challenge for intrusion datasets, eliminating such feature's with spectral information not only improves the classification performance but also helps traditional classifiers to make accurate decision during attack detection, especially when handling with high dimensional and heterogeneous datasets. The NSL_KDD contains sheer volume of network connection data that has posed a continuous challenge to intrusion detection, the growing concern and computational complexities leads to unsatisfactory results. Building classification model on such datasets degrades the performance of the traditional classifiers, and also leads to an insufficient memory storage issues. To solve all the aforementioned problems, the proposed method utilized feature selection technique i.e., QRA, to reduce features as shown in fig. (6). subsequently, the lower dimensional sub-space is used in the training and testing phase, it also helps to reduce proposed method classifier building time and memory issue problem. To analyze the efficiency of proposed method and other state of art ECOC approaches are evaluated on benchmark NSL_KDD dataset. This dataset include multi-class imbalance problem for each of the four attack categories shown in fig. (5), the boundaries for these four categories is difficult to classify majority and minority classes.

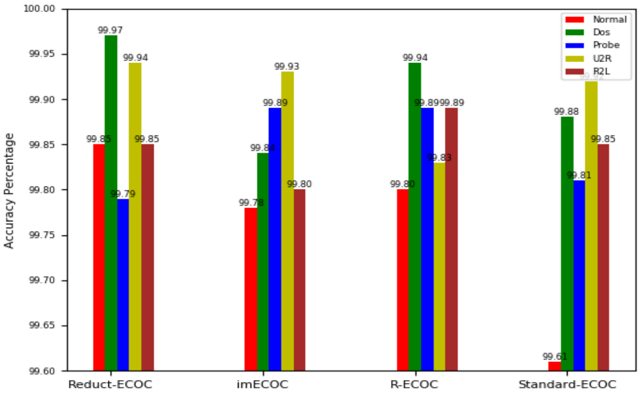


Fig.8: Accuracy measures on NSL_KDD Dataset

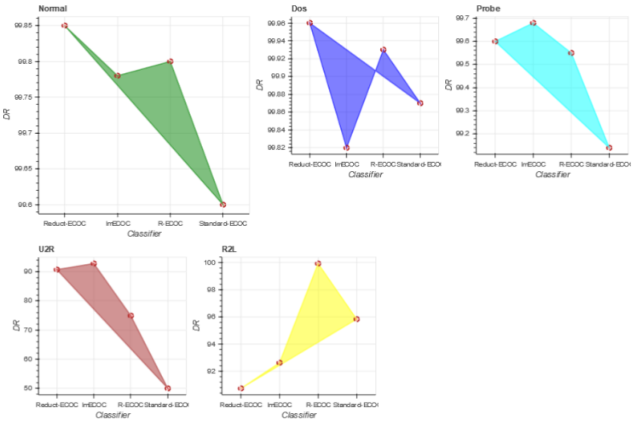


Fig.9: Detection Rate measures on NSL_KDD Dataset

The accuracy measure shown in fig. (8), can be concluded that, standard ECOC method were efficient for the class belonging to the U2R (minority) category, where as other ECOC classifiers failed to identify U2R attacks. Reduct-ECOC identified majority class belonging to Dos and Normal category, while imECOC and R-ECOC achieved best accuracy in detecting class belonging to Probe (majority class) attack. Whereas R-ECOC outperforms in detecting R2L (minority class) attack. The code-matrix defined in R-ECOC is greatly overlooked minority classes, since the minority classes are encoded as positive class, whereas the other classes are encoded as negative class for a dichotomizers. Similar performance may be observed in Fig. (9), the proposed method is good at detecting Normal, Dos and U2R category. imEcoc achieved best DR for U2R category, whereas R-ECOC method is for R2L category. The improvement of U2R category in case of imECOC method is by obtaining optimal dichotomy weights in favor of minority classes. The true positive (TP), false positive (FP), true negative (TN) and false negative (FN) values obtained by Reduct-ECOC classifier on different reduct subspaces with respect to four categories of attacks obtained variations in evaluation measures when compared to other ECOC classifiers are summarized in fig. (6) and fig. (7).The ensemble decisions from multiple reduct-ECOC classifier, C 4.5 as a base learner on different reduct subspaces significantly improved the performance of DR, Accuracy, AUC measures in detecting Normal, Dos and U2R attack categories shown in fig. (8), fig. (9) and fig. (11).

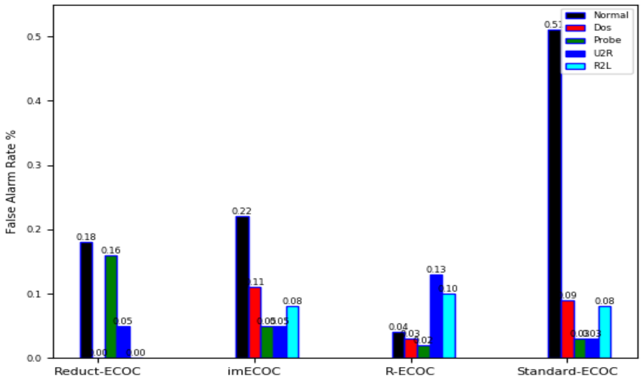


Fig.10: False Alarm Rate on NSL_KDD Dataset

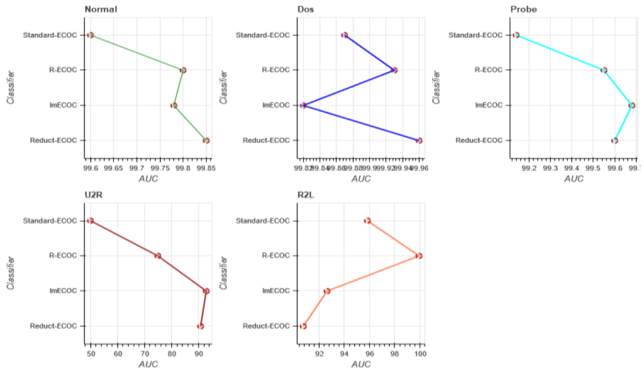


Fig. 11: AUC measures on NSL_KDD Dataset

On examining Fig. (10), the dense random coding strategy shown in fig. (3), reduced FAR for proposed method in the case of Dos and R2L category, whereas R-ECOC achieved lowest FAR for normal category, highest FAR is obtained by standard ECOC method. On the other hand, R-ECOC achieved lowest FAR in all categories when compared to other ECOC classifiers. The average DR w.r.t TP and FP for all ECOC-methods can be found in Table. (2),from the AUC measure, it is clearly observed that, the diversity among dichotomies w.r.t coding strategies may varies from one ECOC-classifier to others, similarly w.r.t hamming distance and Euclidean distance. The code matrix defined for classes should be well separated in terms of rows and columns otherwise it leads to higher FAR and reduces AUC measures.

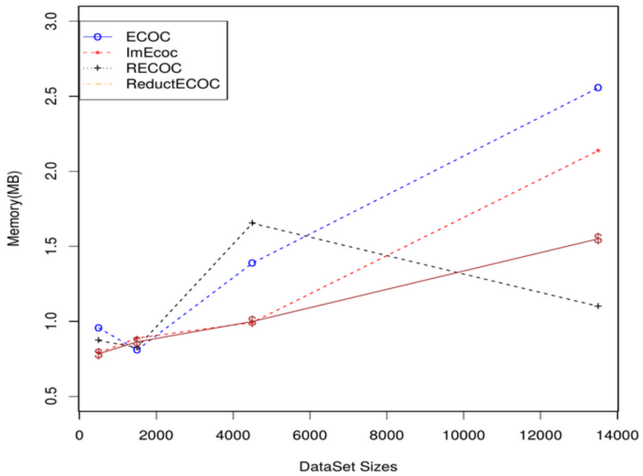


Fig.12: Memory Variance in NSL_KDD

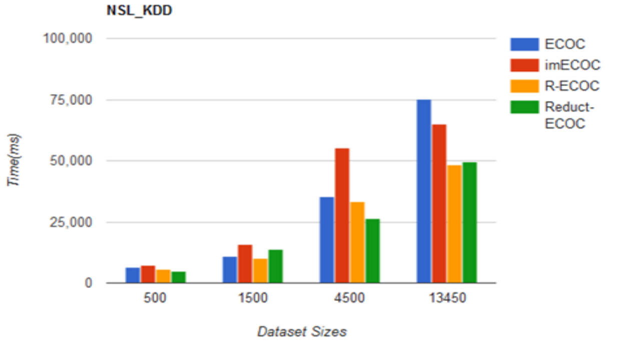


Fig.13: Time Variance in NSL_KDD dataset

Fig. (12) & Fig. (13), show the memory and time taken for building all ECOC classifiers by changing number of training samples. It is clearly observed that, standard ECOC needs higher memory and execution time compared to other ECOC classifiers. The coding strategies shown in fig. 3(a) to fig. 3(d),defined on different sub-partitions of multiple classes w.r.t dichotomizers corresponds to different subspaces may changes both training and testing time.

Table 2: State-of-art ECOC classifier Performance on UCI Datasets

Measures	Datasets	Standard-ECOC	imECOC	R-ECOC	Reduct_ECOC
Accuracy	Iris	0.9911	0.991	0.977	0.9869
	Car	0.9464	0.949	0.864	0.9707
	Balance Scale	0.9990	0.9990	0.9536	0.9855
	Glass	0.8381	0.856	0.863	0.9812
	Wine	0.9764	0.980	0.993	0.9956
	Zoo	1	1	0.979	0.9902
DR	Iris	0.9866	0.986	0.965	0.9810
	Car	0.9	0.907	0.820	0.9484
	Balance Scale	0.9979	0.9979	0.9286	0.9792
	Glass	0.6463	0.695	0.633	0.9358
	Wine	0.9662	0.971	0.99	0.9934
	Zoo	1	1	0.891	0.9604
Precision	Iris	0.9871	0.987	0.965	0.9828
	Car	0.8940	0.901	0.812	0.9467
	Balance Scale	0.9979	0.9979	0.9315	0.9797
	Glass	0.7659	0.816	0.596	0.8982
	Wine	0.9680	0.972	0.990	0.9937

	Zoo	1	1	0.8495	0.9631
	Iris	0.0066	0.0066	0.0169	0.0079
	Car	0.0875	0.0950	0.3416	0.0522
	Balance Scale	0.0011	0.0011	0.0520	0.0196
	Glass	0.0524	0.0454	0.1049	0.0153
	Wine	0.0145	0.0141	0.0038	0.0023
	Iris	0	0	0.0150	0.0085
F-Score	Iris	0.9866	0.9866	0.9650	0.9815
	Car	0.8957	0.8972	0.7920	0.9460
	Balance Scale	0.9979	0.9979	0.9286	0.9791
	Glass	0.6461	0.7055	0.6015	0.9132
	Wine	0.9660	0.9718	0.99	0.9935
	Zoo	1	1	0.8570	0.9562
G-Mean	Iris	0.9867	0.9867	0.9650	0.9817
	Car	0.8963	0.9007	0.8036	0.9468
	Balance Scale	0.9979	0.9979	0.9293	0.9793
	Glass	0.6746	0.7289	0.6081	0.9151
	Wine	0.9666	0.9720	0.9901	0.9935
	Zoo	1	1	0.8634	0.9589

Even though, accuracy measure is not a correct measure to asses model performance when there is a class imbalance problem. Hence, F_Score and G_mean are best evaluation measures to study the class imbalance problem. The results indicated in table2 are best for the ECOC schemes. The Reduct-ECOC achieved best F Score and G-mean values for Car, Glass and Wine dataset, whereas standard-ECOC and imECOC achieved best results for Iris, balance scale and zoo dataset. The F score and G mean measures are most sensitive to detect minority classes, high scores for F-measure and G-mean for all the considered datasets indicate the affinity of the proposed reduct-ECOC approach towards imbalanced datasets. Table 3 depicts the evaluation measures obtained on the iris, car, balance, glass, wine and zoo dataset can be clearly seen that standard ECOC ignores the importance of minority class detection. The F Score and G-mean improvement for most of the dataset in the case of imECOC and reduct-ECOC clearly indicated that weighted decoding mechanism for imbalanced dataset improved the overall detection rate of minority classes.

V.CONCLUSION

This paper presented a novel Reduct-ECOC classification scheme for intrusion detection to deal with multi class or imbalance classification problem. The proposed framework learns on different reduct subspaces and generates multiple independent dichotomy classifiers. Each individual dichotomizer is trained on different reduct feature subsets computed using QuickReduct algorithm. The Reduct-ECOC shows improved performance for both balanced as well as imbalanced datasets. The independent dichotomy classifiers increased the overall accuracy, detection rate and reduced FAR for the overall ensemble classifier. The suitability of proposed ensemble classifier empirically validated on both intrusion detection domain as well as other UCI-ML repository datasets, significantly improved the performance of considered evaluation measures in detecting both known and unknown attack signatures compared to standard ECOC, imECOC, and RSS-ECOC methods.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Science and Engineering Research Board (SERB), Ministry of Science & Technology, Govt. of India under grant No. SB/FTP/ETA -0180/2014 for providing partial funding support.

REFERENCES

1. Rodda, S., & Erothi, U. S. A Roughset Based Ensemble Framework for Network Intrusion Detection System. International Journal of Rough Sets and Data Analysis (IJRSDA), 5(3):71-88, 2018.
2. Rodda, S., & Erothi, U. S. R. Network Intrusion Detection System to Preserve User Privacy. In Proceedings of International Conference on Computational Intelligence and Data Engineering. Springer, Singapore, pages 85-94, 2018.
3. Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., ... & Zissman, M. A. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. DARPA Information Survivability Conference and Exposition, IEEE, 2: 12-26, 2000.
4. Tavallaee, M., Bagheri, E., Lu, W., &Ghorbani, A. A. A detailed analysis of the KDD CUP 99.
5. S.Revathi, Dr A.Malathi. A detailed analysis on NSL-KDD data set using various machine learning techniques for intrusion detection,2(12):1848-1853, 2013.
6. Rodda, S., & Erothi, U. S. R. Class Imbalance Problem in the Network Intrusion Detection Systems. In Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on. IEEE, 2016, pages 2685-2688, 2016.
7. Bengio, S., Weston, J., & Grangier, D. Label embedding trees for large multi-class tasks. In Advances in Neural Information Processing Systems, pages 163-171, 2010.
8. Crammer, K., Singer, Y, on the algorithmic implementation of multiclass kernel-based vector machines. Journal of machine learning research, 2:265–292, 2002.
9. Schapire, Robert E. A brief introduction to boosting. Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 2:1401-1406, 1999.
10. Breiman, Leo. Random forests. Machine learning, 45(1):5-32, 2001.
11. T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. Journal of artificial intelligence research, pages 263–286, 2012.
12. Gholam Ali Montazer, Sergio Escalera, et al. Error correcting output codes for multiclass classification: Application to two image vision problems. In AISP, pages 508–513, 2012.
13. A. Fernandez, V. Lopez, M. Galar, M. Jose del Jesus, F. Herrera. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. Knowledgebase Systems, 42:97-110, 2013.
14. E. L. Allwein, R. E. Schapire, Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research, 1:113–141, 2001.
15. X. Baro, S. Escalera, J. Vitria, O. Pujol, P. Radeva. Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification. IEEE Transactions on Intelligent Transportation Systems, 10(1):113–126, 2009.
16. OriolPujol, PetiaRadeva, Jordi Vitria. Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes. In IEEE Transaction on Pattern Analysis and Machine Intelligence, pages 1007–1012, 2016.
17. Sergio Escalera ,OriolPujol. Ecoc-one: A novel coding and decoding strategy. In ICPR, pages 578–581, 2006.
18. GuoqiangZhong,YuchenZheng,Peng Zhang,Mengqi Li,JunyuDong. DEEP Error Correcting Output Codes. Under review as a conference paper at ICLR, pages 1-11, 2017.

19. Liu, Xu-Ying, Qian-Qian Li, Zhi-Hua Zhou. Learning imbalanced multi-class data with optimal dichotomy weights. In Data Mining (ICDM), IEEE 13th International Conference on Data Mining, pages 478-487, 2013.
20. X. Wang ,Q. Ji. Video Event Recognition with Deep Hierarchical Context Model. In CVPR, pages 4418–4427, 2015.
21. Kittler, Josef, et al. on combining classifiers. IEEE transactions on pattern analysis and machine intelligence, 20(3): 226-239, 1998.
22. Ghani, Rayid. Combining labeled and unlabeled data for text classification with a large number of categories. ICDM 2001, Proceedings IEEE International Conference on Data Mining, 2001.
23. Bagheri, Mohammad Ali, Qigang Gao, and Sergio Escalera. Rough set subspace error-correcting output codes. Data Mining (ICDM), 2012 IEEE 12th International Conference on. IEEE, pages 822-827, 2012.
24. Hsu, Chih-Wei, and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. IEEE transactions on Neural Networks, 13(2): 415-425, 2002.
25. Pawlak, Zdzisław. Rough set theory and its applications. Journal of telecommunications and information technology, pages 7-10, 2002.
26. Chouchoulas, Alexios, and Qiang Shen. Rough set-aided keyword reduction for text categorization. Applied Artificial Intelligence, 15(9):843-873, 2001.
27. Blake, C., &Merz, C. J, {UCI} Repository of machine learning databases, 1998.
28. Erothi, U. S. R., & Rodda, S. Fuzzy ECOC Framework for Network Intrusion Detection System.

AUTHORS PROFILE

Mr. Uma Shankar Rao Erothi, is currently an Assistant Professor in Department of CSE at RAGHU Institute of Technology, Visakhapatnam, Andhra Pradesh, India. He received his M.Tech (CSE) degree in the year 2013.He received B.Tech (CSE) degree in the year 2007. He published more than 7 papers in referred national and international journals. His research interest Data mining, Image Processing and Data Structures.

Dr. Sireesha Rodda, is a Professor in Department of Computer Science and Engineering at GITAM University, Visakhapatnam, India. She has published more than 25 papers in refereed National and International Journals. Her research interests include machine learning and big data analytics.

Retrieval Number: B4238129219/2020©BEIESP
DOI: 10.35940/ijeat.B4238.029320
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication