Expert Systems With Applications 213 (2023) 118929

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Improvement of accuracy of under-performing classifier in decision making using discrete memoryless channel model and Particle Swarm Optimization

Rajasekharreddy Poreddy *, E.S. Gopi

Pattern Recognition and Computational Intelligence Laboratory, Department of Electronics and Communication Engineering, National Institute of Technology Tiruchirappalli, 620015, Tamil Nadu, India

ARTICLE INFO

Keywords:
Bayes cost
Randomized decision rule
Mini-max technique
Particle Swarm Optimization
Classification
Decision making

ABSTRACT

In-spite of availability of wide range of algorithms for constructing multi-class classifier, there are applications like Decision support system, Game prediction in the sports, where the multi-class classifier performs relatively poor in terms of achieving the reasonable classification accuracy. In this paper, poorly performing multi-class classifier (constructed using the classical methods like Artificial Neural Network, cascade of Support Vector Machine, etc.) is treated as the discrete memoryless channel model with known transition probabilities (channel matrix) and the unknown priors. It is further used to construct M-ary Mini-Max technique based randomized decision rule to improve the performance of the multi-class classifier in terms of the classification accuracy. The prior probabilities and the probabilities associated with the M-ary randomized decision rule are further solved using the Particle Swarm Optimization. The experimental results based on the Monte-Carlo simulation using the synthetic data set and the real data set reveal the consistent improvement in the performance of the poorly performing classifier using the proposed technique.

1. Introduction

Humans must make decisions in every situation in their daily lives. These decisions can be made based on the information available in the situation or on previous experience. As a result, decision-making can be divided into two categories. They are *description based-decision making* and *experience based-decision making* (Kudryavtsev & Pavlodsky, 2012). The information about the outcomes and the respective possibilities is specified in description-based decision making, whereas in experience-based decision making, the decision maker must learn the possibilities and the respective outcomes through observation. The decision-making process will be improved by incorporating experience in description-based decision making (Newell & Rakow, 2007). Decision making process can used in different applications such as recommendation systems (Barbosa, 2012; Chai et al., 2021; Chen et al., 2013), medical applications (Leberia et al., 2022; Shamout et al., 2021), education applications (Muhammad et al., 2021; Yuri et al., 2019), business applications (Felipe et al., 2019; Marco et al., 2022), household applications (Karla et al., 2021), agricultural applications (Mostafa & Mozaffar, 2022), cyber security (Chale & Bastian, 2022) etc.,

These days, machine learning (ML) algorithms are popular methods for data analysis. These algorithms have outperformed pure statistics techniques in this regard (Rodriguez et al., 2015). Artificial neural networks (ANN), support vector machines (SVM), Naive Bayes (NB), and other state-of-the art classification algorithms are widely used in decision-making applications. These classification techniques may not be effective in making the final decision in some applications (Felipe et al., 2019; Hallajian et al., 2022; Krishna et al., 2022; Mostafa & Mozaffar, 2022; Singh & Biswas, 2022; Yu-Lin et al., 2022). In Felipe et al. (2019), for example, SVM was not very effective in predicting day-to-day trade investments on the stock market for decision making. In Hallajian et al. (2022), Table 5 presents that the classifiers SVM and NB have under-performed on some datasets. The NB classifier and SVM have given a satisfactory prediction results on Vehicle, WineQR and WineQW datasets in Yu-Lin et al. (2022). In Haung et al. (2014), unsupervised Extreme Learning Machine (US-ELM) algorithm was not able to give good accuracy on the YALEB dataset. The classifiers such as SVM, NB, KNN and Random Forest (RF) were not able to classify the under water acoustic dataset using the features like Mel-spectrogram, MFCC and wavelets in Irfan et al. (2021).

The classification performance for decision making can be improved using different techniques like feature extraction (Irfan et al., 2021; Singh & Biswas, 2022), feature selection (Hallajian et al., 2022; Mostafa & Mozaffar, 2022), increasing number of data samples to train the classifiers (Ahuja & Vishwakarma, 2021), Using different classifiers, training the classifiers for more time etc.,. However, when the dataset is having less samples and limited number of attributes, then the

* Corresponding author.
E-mail addresses: sekharpraja@gmail.com (R. Poreddy), esgopi@nitt.edu (E.S. Gopi).

https://doi.org/10.1016/j.eswa.2022.118929
Received 21 April 2022; Received in revised form 8 September 2022; Accepted 25 September 2022
Available online 30 September 2022

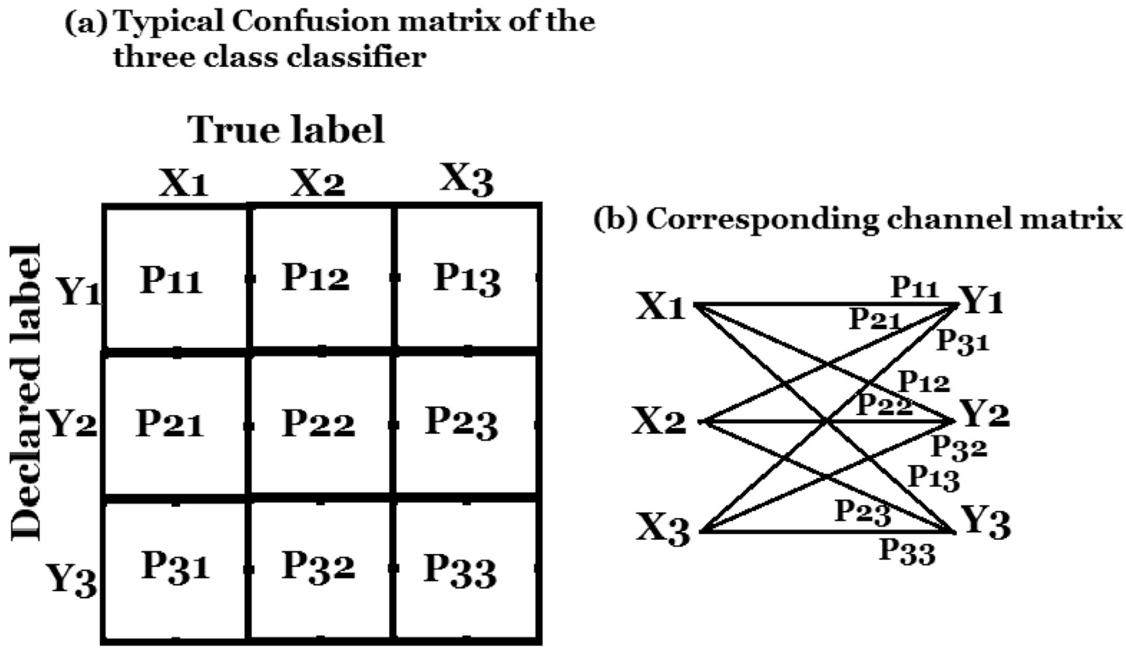

Fig. 1. Illustration of the construction of DMC using the confusion matrix of the trained M-class classifier.

feature extraction and selection methods may not work in improving the classification accuracy. In such cases where the classifier is under-performing, this paper employs a novel approach to improve the performance of classification algorithms for decision making by using the predicted results of the classifier.

The proposed methodology can be used in the following applications, (1) to improve the performance of the classifier when there are limited options, such as improving the dataset (taking more samples or data augmentation etc.,) or refining the classification technique, and (2) to aid human decision making when the human must choose between multiple options based on the past experience. The proposed methodology takes into account human decision-making history as well as the actual choice that should have been made in order to suggest the best option for a new situation.

The main contributions of this paper are

1. Constructing Discrete Memoryless Channel (DMC) Model using the trained multi-class classifier's confusion matrix and improving the classification accuracy using Particle Swarm Optimization (PSO) based M-ary mini-max Hypothesis Testing.
2. Demonstrating the effectiveness of the proposed technique when applied on the prediction of the classifier or predictor using the synthetic and real data set.
3. Demonstrating the impact of the proposed technique in improving the human decision making using a sports application.

The rest of the paper is organized as follows. Section 2 describes the problem formulation and proposed methodology to improve the detection rate of the under-performing classifier. Section 3 describes proposed solution for mini-max testing problem using PSO. Section 4 studies the performance of the proposed technique on synthetic and real datasets. Finally, Section 5 concludes this paper.

2. Problem formulation and the proposed methodology

This section explains the construction of the DMC using the confusion matrix of the classifier. Consider the DMC model for data transmission which consists of N number of the transmitter symbols and N number of received symbols. Let X_i be the ith transmitted symbol from the transmitter and the corresponding received symbol is Y_i. The detection of symbol X_i based on the received symbol Y_i involves partitioning the N symbols into M sets namely $\Gamma_1, \Gamma_2, \ldots \Gamma_M$. If the received symbol Y_i lies in the set Γ_j, declare the transmitted symbol as X_j. The partition set is obtained by minimizing the Bayes cost, which depends on the prior probabilities of the transmitting symbols. In case of unknown prior probabilities, Mini-max technique is used for identifying the prior probability ($\boldsymbol{\pi}$) at which maximum of the conditional costs are minimized. In discrete case, decision rule corresponding to $\boldsymbol{\pi}$ is the randomized decision rule, in which different partition sets are used with different probabilities. Thus Mini-max based randomized decision rule is attained given the channel matrix of the DMC.

In this proposal, We mimic trained M-class classifier as the DMC with $M = N$. True labels are considered as the sequence of data being transmitted from the transmitter and the corresponding detected labels (by the trained classifier) are treated as the sequence of symbols received in the receiver. The confusion matrix attained using the trained M-class classifier with the training set is used to formulate the channel matrix of the typical DMC with $M = N$ (refer Fig. 1). The Mini-Max technique based randomized decision rule is further obtained for the constructed DMC. This completes the training phase. The trained M-Class classifier cascaded with the Randomized decision rule is used to classify the sample under test (refer Fig. 2).

DMC model with the typical channel matrix attained using the confusion matrix of the classifier is considered as M-ary Hypothesis testing problem with unknown prior probabilities as described in Section 2.2.

2.1. Notations

Throughout the manuscript, the vectors are denoted by bold lowercase letters and the matrices are denoted by bold uppercase letters. The partition set and the observation space are denoted by Γ and $\mathscr{F}$, respectively. All the letters which are not in bold denote scalars.

2.2. M-ary hypothesis testing

M-ary hypothesis testing problem consists of the hypotheses $H_j, j = 1, 2, \ldots, M$, in the observation space $(\Gamma, \mathscr{F})$, are defined as

$$H_j : y = j, j = 1, 2, \ldots, M \tag{1}$$

where the observation set $\Gamma = \{1, 2, \ldots, M\}$ and $\mathscr{F}$ is the σ-Field of Γ.

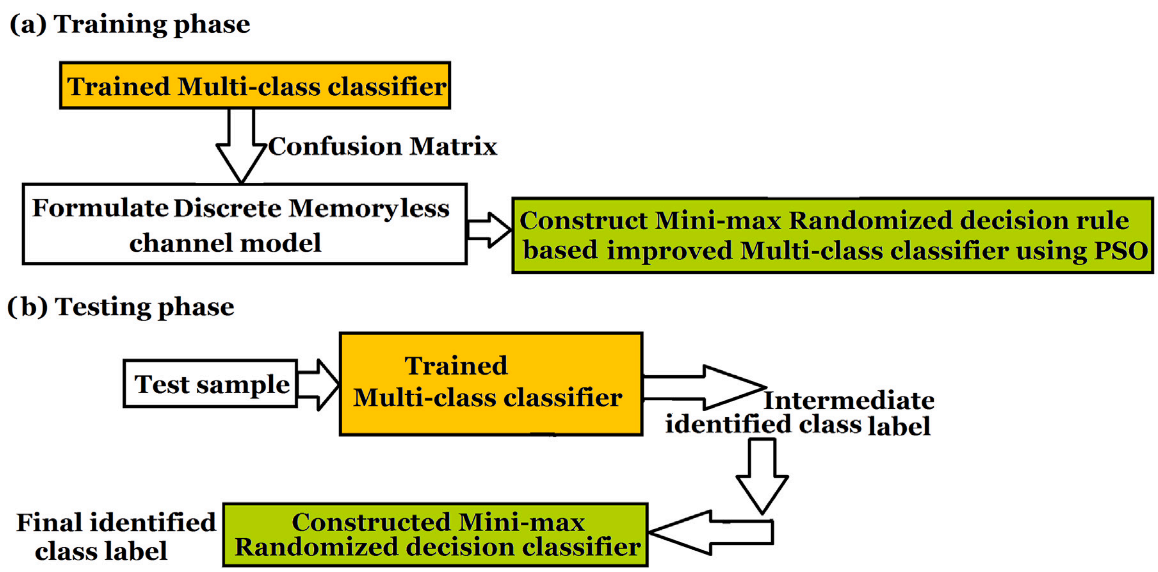

Fig. 2. Block diagram of the proposed model.

The decision rule δ involves in partitioning the observation space Γ into M disjoint sets $\Gamma_i \in \mathscr{F}$, $i = 1, 2, \dots, M$, such that if $y \in \Gamma_i$, decide the outcome as i. i.e.,

$$\delta(y) = i, \text{if } y \in \Gamma_i \tag{2}$$

for $i = 1, 2, \dots, M$. The decision rule will change based on the partition of the observation space Γ. The number of distinct possible partitions of the observation space for M hypotheses are M^M (refer Appendix). There is a chance of choosing the hypothesis H_i when the hypothesis H_j is true. Hence the cost of choosing the hypothesis H_i when the hypothesis H_j is true is considered as C_{ij} and the conditional cost of the decision rule δ for the true hypothesis H_j is calculated as $B_j(\delta) = \sum_{\substack{i=1 \\ i \neq j}}^{M} C_{ij} P_j(\Gamma_i)$. Most commonly used cost assignment is the uniform cost assignment (Poor, 1994) which is given by $C_{ij} = \begin{cases} 1, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}$ Hence, the conditional costs become $B_j(\delta) = \sum_{\substack{i=1 \\ i \neq j}}^{M} P_j(\Gamma_i)$ and the overall average cost is formulated as

$$c(\boldsymbol{\pi}, \delta) = \sum_{j=1}^{M} \sum_{\substack{i=1 \\ i \neq j}}^{M} \pi_j P_j(\Gamma_i) \tag{3}$$

where $\boldsymbol{\pi} = [\pi_1, \pi_2, \dots, \pi_M]^T$, π_j is the prior probability of the hypothesis H_j and $\sum_{i=1}^{M} \pi_j = 1$. The decision rule δ which minimizes the overall average cost (Bayes rule) needs to be obtained for the known prior probabilities. Since the prior probabilities are assumed to be not known in the formulated DMC model, Mini-Max hypothesis testing is used to obtain the optimal decision rule as described below.

The aim of the mini-max hypothesis testing is to minimize the maximum of the conditional cost, $B_j(\delta), \forall j$, over all possible decision rules (Poor, 1994). The problem of $max B_j(\delta), \forall j$, for a decision rule δ is nothing but $max c(\boldsymbol{\pi}, \delta), \forall \boldsymbol{\pi}$. Hence, the optimization problem for mini-max technique can be formulated as

$$\hat{\boldsymbol{\pi}} = arg \min_{\delta} \max_{\boldsymbol{\pi}} c(\boldsymbol{\pi}, \delta) \tag{4}$$

This problem is reformulated by considering the symmetry property mentioned in Poor (1994) and the Bayes cost $V(\boldsymbol{\pi}) = \min_{\delta} c(\boldsymbol{\pi}, \delta)$ as

$$\hat{\boldsymbol{\pi}} = arg \max_{\boldsymbol{\pi}} V(\boldsymbol{\pi}) \tag{5}$$

This problem can be solved only when the Bayes rule $\delta_{\boldsymbol{\pi}}$ and the Bayes cost $V(\boldsymbol{\pi})$ are known at the prior probability $\boldsymbol{\pi}$.

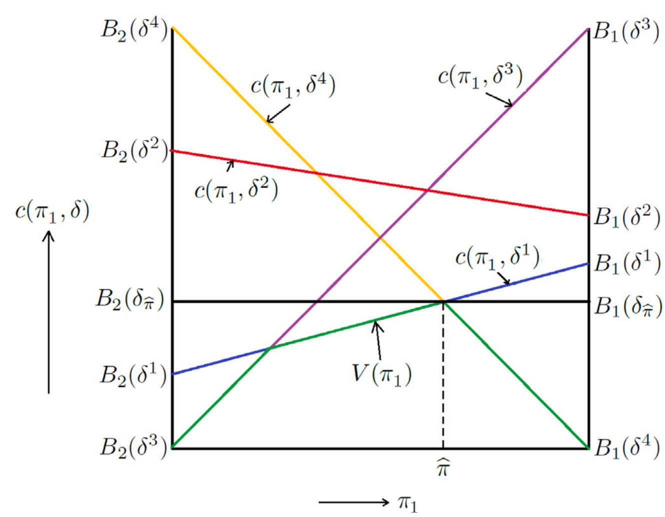

Fig. 3. Illustrations of the Bayes cost, $V(\pi_1)$ (green curve) for binary channel model.

2.2.1. Binary channel model

The number of distinct partitions of the observation set Γ for binary channel model are 4. They are (1) $\Gamma_1 = \{1\}$, $\Gamma_2 = \{2\}$, (2) $\Gamma_1 = \{2\}$, $\Gamma_2 = \{1\}$, (3) $\Gamma_1 = \{1, 2\}$, $\Gamma_2 = \emptyset$, and (4) $\Gamma_1 = \emptyset$, $\Gamma_2 = \{1, 2\}$. Hence, four different decision rules can used to make the final decision. The typical overall average cost $c(\pi_1, \delta)$ for all the four decision rules and the Bayes cost $V(\pi_1)$ (green curve) are illustrated as a function of the prior probability π_1 in Fig. 3 (Gopi, 2016). The maximum of $V(\pi_1)$ is obtained at $\hat{\pi}$ where two decision rules (δ_1 and δ_3) are having equal Bayes cost $V(\pi_1)$. The final decision on the received symbol is made by randomly selecting one of the two Bayes rules at $\hat{\pi}$ using randomized decision rule. The probability of selecting a Bayes rules, δ_1 or δ_4 is obtained by equating the conditional costs, $B_1(\delta)$ and $B2(\delta)$ of both the decision rules.

Finding the Bayes cost $V(\boldsymbol{\pi})$ for a particular prior probability $\boldsymbol{\pi}$ by evaluating all the decision rules δ is computationally cost inefficient for the cases when $M > 2$, as the number of decision rules increase exponentially with M. Hence a novel methodology is proposed using the joint probability matrix and PSO to solve the optimization problem in (5). Once the prior probabilities, $\hat{\boldsymbol{\pi}}$ and the corresponding Bayes rules $\delta_{\hat{\boldsymbol{\pi}}}$ are computed, the corresponding outcome can be decided using the *randomized decision rule* as mentioned in Section 3.3 by finding the probabilities of the Bayes rules to make the final decision using PSO.

3. Mini-max hypothesis testing using PSO

In order to solve the M-ary Mini-max hypothesis testing problem formulated in (5) using PSO, let us first consider the probability of correct decision of all classes for the prior probability $\boldsymbol{\pi}$ and the decision rule δ be defined as

$$p_c(\boldsymbol{\pi}, \delta) = \sum_{i=1}^{M} \pi_i P_i(\Gamma_i) \tag{6}$$

then the Bayes cost $V(\boldsymbol{\pi})$ can be formulated as

$$V(\boldsymbol{\pi}) = 1 - \max_{\delta} p_c(\boldsymbol{\pi}, \delta) \tag{7}$$

Using (6) and (7), the mini-max problem is reformulated as

$$\begin{aligned} \hat{\boldsymbol{\pi}} &= arg \max_{\boldsymbol{\pi}} (1 - \max_{\delta} p_c(\boldsymbol{\pi}, \delta)) \\ &= arg \min_{\boldsymbol{\pi}} \max_{\delta} p_c(\boldsymbol{\pi}, \delta) \end{aligned} \tag{8}$$

Solving (8) involves identifying the prior probability $\hat{\boldsymbol{\pi}}$ that minimizes $p_c^{max}(\boldsymbol{\pi})$ for all prior probabilities $\boldsymbol{\pi}$. i.e.,

$$\hat{\boldsymbol{\pi}} = arg \min_{\boldsymbol{\pi}} p_c^{max}(\boldsymbol{\pi}). \tag{9}$$

In this $p_c^{max}(\boldsymbol{\pi})$ is obtained by maximizing $p_c(\boldsymbol{\pi}, \delta)$ for all the decision rules δ at a particular prior probability $\boldsymbol{\pi}$. The corresponding decision rule $\delta_{\boldsymbol{\pi}}$ is known as Bayes rule (s).

$$\begin{aligned} p_c^{max}(\boldsymbol{\pi}) &= \max_{\delta} p_c(\boldsymbol{\pi}, \delta) \\ \delta_{\boldsymbol{\pi}} &= arg \max_{\delta} p_c(\boldsymbol{\pi}, \delta) \end{aligned} \tag{10}$$

If more than one Bayes rule gives the identical maximum value of $p_c(\boldsymbol{\pi}, \delta)$ for the particular prior probability $\boldsymbol{\pi}$, randomized decision rule is obtained (refer 3.3). The prior probabilities and the probabilities used in the randomized decision rule are optimized using PSO as described in 3.2 and 3.3 respectively.

3.1. Bayes rule(s) at prior probabilities $\boldsymbol{\pi}$

Given the elements of the channel matrix and prior probabilities, the matrix $\mathbf{S}$ is constructed as follows.

$$\mathbf{S} = \begin{bmatrix} \pi_1 P(1|1) & \pi_1 P(2|1) & \cdots & \pi_1 P(M|1) \\ \pi_2 P(1|2) & \pi_2 P(2|2) & \cdots & \pi_2 P(M|2) \\ & & \vdots & \\ \pi_M P(1|M) & \pi_M P(2|M) & \cdots & \pi_M P(M|M) \end{bmatrix} \tag{11}$$

where $P(j/i)$ is the (i,j)th element of the channel matrix and π_i corresponds to the prior probability of the ith transmitting symbol. For the given prior probabilities, it is observed that the maximum value of the probability of correct decision is obtained as $p_c^{max}(\boldsymbol{\pi}) = \sum_{j=1}^{M} \max_i s_{ij}$, where s_{ij} is the element in the ith row and jth column of the matrix $\mathbf{S}$. The corresponding decision rule (Bayes rule(s) $\delta_{\boldsymbol{\pi}}$ at the prior $\boldsymbol{\pi}$), is determined by assigning j to the partition set Γ_i, if the column j of the matrix $\mathbf{S}$ is having the maximum at row i. If more than one element of the jth column is maximum, one among the elements is chosen to formulate the typical Bayes rule. This is repeated by choosing other maximum elements one after another to obtain multiple Bayes rules. For the given matrix S, the Bayes rule(s) are obtained as illustrated in Fig. 4. The values in the red boxes are the maximum values in the respective columns. It is observed that two Bayes rules are obtained when two elements of the typical column of the matrix S are maximum. It is observed that Γ_i, with $i = 1, 2, 3$ forms the partition sets in all the decision rules.

3.2. Prior probabilities ($\hat{\boldsymbol{\pi}}$) using PSO

PSO is a behaviorally inspired evolutionary computational intelligence algorithm for optimization problems. PSO is based on the behavior of flock of birds to choose the optimum (shortest) path to reach the destination (Gopi, 2007). This is used to minimize the arbitrary objective function $J(x)$, where x and $J(x)$ are considered as the position of the bird and the corresponding distance from the destination.

The minimization problem in (9) is solved using the PSO. The particles of the swarm are the prior probabilities and the dimension of the particle depends upon the number of hypotheses M. These particles of the swarm are generated randomly with uniform distribution in $[0, 1]$ and normalized to make sure the sum of the particle is one. The functional value of the objective function $p_c^{max}(\boldsymbol{\pi})$ is determined for each generated particle as explained in Section 3.1. The prior probability (particle) that gives the minimum functional value is declared as global best.

Algorithm 1 PSO for finding the prior probabilities ($\hat{\boldsymbol{\pi}}$)

Initialization: Assume the number of particles in the swarm as L, the dimension of each particle is M, the number of iterations K, c_1 and c_2 are the individual and group learning rates, r_1 and r_2 are uniformly distributed random numbers between 0 and 1,
$|.|$ denotes the modulus and
the minimum value of the objective function p_c^{max} is saved in the variable **min_obj1** to track the convergence of the algorithm.
Initialize the L particle positions randomly as $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_L$ and the tentative local decisions taken by the particles randomly as $\mathbf{ld}_1, \mathbf{ld}_2, \ldots, \mathbf{ld}_L$.
for $k = 1$ to K
Step 1: Compute the functional values of $p_c^{max}(\mathbf{ld}_i), \forall i$ as explained in Section 3.1
Step 2: Find the minimum among $p_c^{max}(\mathbf{ld}_i), \forall i$ and declare the particle position as global best, $\mathbf{ld}_g$. i.e., $\mathbf{ld}_g = \arg\min_{i=1,\ldots,L} p_c^{max}(\mathbf{ld}_i)$ and $p_c^{max}(\mathbf{ld}_g) = \min_{i=1,\ldots,L} p_c^{max}(\mathbf{ld}_i)$.
Step 3: Identify the actually moved next position of the particles.
  for $i = 1$ to L
    $\mathbf{next\_v}_i = |\mathbf{v}_i + c_1 r_1(\mathbf{ld}_i - \mathbf{v}_i) + c_2 r_2(\mathbf{ld}_g - \mathbf{v}_i)|$
    $\mathbf{next\_v}_i = \mathbf{next\_v}_i / sum(\mathbf{next\_v}_i)$
  end for
Step 4: Assign the current positions of the particles as $\mathbf{v}_i = \mathbf{next\_v}_i, \forall i$ and the tentative decision taken by each particle for further movement as
  for $i = 1$ to L
    if $p_c^{max}(\mathbf{next\_v}_i) < p_c^{max}(\mathbf{ld}_i)$
      $\mathbf{ld}_i = \mathbf{next\_v}_i$
    else
      $\mathbf{ld}_i = \mathbf{ld}_i$
    end if
  end for
min_obj1$_k = \min_{i=1,\ldots,L} p_c^{max}(\mathbf{v}_i)$
  if $(\mathbf{min\_obj1}_{k-1} - \mathbf{min\_obj1}_k) < 10^{-6}$
    Break the loop and go to Step5
  end if
end for
Step5: Compute the functional values $p_c^{max}(\mathbf{v}_i)$ and find the $\hat{\boldsymbol{\pi}} = \arg\min_{i=1,\ldots,L} p_c^{max}(\mathbf{v}_i)$

The next positions of prior probabilities are obtained by following step 5 of PSO algorithm and these positions may not satisfy the following axioms of the probability (Papoulis & Pillai, 2002). (1) $\pi_j > 0$, (2) $\sum_{j=1}^{M} \pi_j = 1$. To satisfy the axiom (1), the next position of a particle is

$$\begin{array}{c c c}
\underline{\text{Matrix } \mathbf{S}} & & \begin{array}{c}\delta_{\pi}\\ \underline{\text{Decision Regions}}\end{array} \\
\begin{bmatrix} \pi_1 P(1|1) & \boxed{\pi_1 P(2|1)} & \pi_1 P(3|1) \\ \boxed{\pi_2 P(1|2)} & \pi_2 P(2|2) & \boxed{\pi_2 P(3|2)} \\ \pi_3 P(1|3) & \pi_3 P(2|3) & \pi_3 P(3|3) \end{bmatrix} & \begin{array}{c}\longrightarrow\\ \longrightarrow\\ \longrightarrow\end{array} & \begin{array}{l}\Gamma_1 = \{2\}\\ \Gamma_2 = \{1,3\}\\ \Gamma_3 = \{\}\end{array}
\end{array}$$

(a)

$$\begin{array}{c c c c}
\underline{\text{Matrix } \mathbf{S}} & & \begin{array}{c}\delta_{\pi}^{1}\\ \underline{\text{Decision Regions}}\end{array} & \begin{array}{c}\delta_{\pi}^{2}\\ \underline{\text{Decision Regions}}\end{array} \\
\begin{bmatrix} \pi_1 P(1|1) & \boxed{\pi_1 P(2|1)} & \pi_1 P(3|1) \\ \boxed{\pi_2 P(1|2)} & \pi_2 P(2|2) & \boxed{\pi_2 P(3|2)} \\ \pi_3 P(1|3) & \pi_3 P(2|3) & \boxed{\pi_3 P(3|3)} \end{bmatrix} & \begin{array}{c}\longrightarrow\\ \longrightarrow\\ \longrightarrow\end{array} & \begin{array}{l}\Gamma_1 = \{2\}\\ \Gamma_2 = \{1,3\}\\ \Gamma_3 = \{\}\end{array} & \begin{array}{l}\Gamma_1 = \{2\}\\ \Gamma_2 = \{1\}\\ \Gamma_3 = \{3\}\end{array}
\end{array}$$

(b)

Fig. 4. Illustration of the selection of the Bayes rule δ_{π} at π using the matrix **S** for 3 class problem, when (a) the number of Bayes rules is only one and (b) the number of Bayes rules is two. The values in the red boxes are the maximum values in the respective columns.

obtained by taking the absolute values of the update equation and to satisfy the axiom (2), the particles need to be normalized such that the summation over the elements of each particle becomes one.

The algorithmic framework of the PSO for finding $\hat{\pi}$ is presented in Algorithm 1, where the complexity of the Algorithm 1 is in the order of $O(MLK)$. where M is the number of hypotheses, L is the number particles in the swarm and K is the number of iterations that the PSO algorithm is repeated till the converge.

Once the prior probability $\hat{\pi}$ are found, the decision rules $\delta^1_{\hat{\pi}}, \delta^2_{\hat{\pi}}, \dots, \delta^n_{\hat{\pi}}$, that give the Bayes cost can be obtained using the matrix **S** as explained in Section 3.1. The number n will vary according to the channel matrix (cost matrix) and it ranges from 1 to M i.e., $1 \le n \le M$. In discrete channel model, there will be a discontinuity in the slope of $V(\pi)$ at mini-max cost. Hence the randomized decision rule is used to choose the decision rule for making the final decision on the hypothesis at a particular instance.

3.3. Randomized decision rule

The final decision of choosing the hypothesis is made by the decision rule $\delta^k_{\hat{\pi}}$ with probability ρ_k. This process of choosing a decision rule based on probability is known as *randomized decision rule*. The randomized decision rule is given by

$$\delta_{\hat{\pi}}(\rho) = \begin{cases} \delta^1_{\hat{\pi}}, & \text{if } 0 < \rho \le \rho_1 \\ \delta^2_{\hat{\pi}}, & \text{if } \rho_1 < \rho \le (\rho_1 + \rho_2) \\ \vdots & \vdots \\ \delta^n_{\hat{\pi}}, & \text{if } (\rho_1 + \rho_2 + \cdots + \rho_{n-1}) < \rho \le 1 \end{cases} \tag{12}$$

where ρ is random number ranging from 0 to 1. The final decision on the outcome will be made using the selected Bayes rule as shown in Eq. (2). The probabilities $\rho_k, k = 1, 2, \dots, n$ are calculated by equating the conditional costs of all the classes at $\pi = \hat{\pi}$: $B_1(\delta_{\hat{\pi}}) = B_2(\delta_{\hat{\pi}}) = \cdots = B_M(\delta_{\hat{\pi}})$, where $B_i(\delta_{\hat{\pi}})$ is the conditional cost of class i and is given by $B_i(\delta_{\hat{\pi}}) = \rho_1 B_i(\delta^1_{\hat{\pi}}) + \rho_2 B_i(\delta^2_{\hat{\pi}}) + \cdots + \rho_n B_i(\delta^n_{\hat{\pi}})$ for $i = 1, 2, \dots, M$, and $\sum_{k=1}^{n} \rho_k = 1$.

Hence this problem is solved by making use of the following objective function

$$J(\rho) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \left(B_i(\delta_{\hat{\pi}}) - B_j(\delta_{\hat{\pi}})\right)^2 \tag{13}$$

where $\rho = [\rho_1, \rho_2, \dots, \rho_n]^T$. It is observed that the number of Bayes rules n at $\hat{\pi}$ is not constant and varies based on the channel matrix H. The

---

Algorithm 2 PSO for finding Bayes rule probability (ρ)

Initialization: Assume the number of particles in the swarm is L, the dimension of each particle is n, the number of iterations K, c_1 and c_2 are the individual and group learning rates, r_1 and r_2 are uniformly distributed random numbers between 0 and 1, $|.|$ denotes the modulus and the minimum value of the objective function $J(\rho)$ given in (13) is saved in the variable **min_obj2** to track the convergence of the algorithm.

Initialize the L particle positions randomly as $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L$ and the tentative local decisions taken by the particles randomly as $\mathbf{ld}_1, \mathbf{ld}_2, \dots, \mathbf{ld}_L$.

for $k = 1$ to K

Step 1: Compute the functional values of $J(\mathbf{ld}_i), \forall i$.

Step 2: Find the minimum among $J(\mathbf{ld}_i), \forall i$ and declare the particle position as global best, $\mathbf{ld}_g$. i.e., $\mathbf{ld}_g = \arg\min_{i=1,\dots,L} J(\mathbf{ld}_i)$ and $J(\mathbf{ld}_g) = \min_{i=1,\dots,L} J(\mathbf{ld}_i)$.

Step 3: Identify the actually moved next position of the particles.

  for $i = 1$ to L

    $\mathbf{next\_v}_i = |\mathbf{v}_i + c_1 r_1(\mathbf{ld}_i - \mathbf{v}_i) + c_2 r_2(\mathbf{ld}_g - \mathbf{v}_i)|$

    $\mathbf{next\_v}_i = \mathbf{next\_v}_i / sum(\mathbf{next\_v}_i)$

  end for

Step 4: Describe the current positions of the L particles as $\mathbf{v}_i = \mathbf{next\_v}_i, \forall i$ and the tentative decision taken by each particle for further movement is obtained as

  for $i = 1$ to L

    if $J(\mathbf{next\_v}_i) < J(\mathbf{ld}_i)$

      $\mathbf{ld}_i = \mathbf{next\_v}_i$

    else

      $\mathbf{ld}_i = \mathbf{ld}_i$

    end if

  end for

$\mathbf{min\_obj2}_k = \min_{i=1,\dots,L} J(\mathbf{v}_i)$

  if $(\mathbf{min\_obj2}_{k-1} - \mathbf{min\_obj2}_k) < 10^{-6}$

    Break the loop and go to Step5

  end if

end for

Step5: Compute the functional values $J(\mathbf{v}_i)$ and find the $\rho = \arg\min_{i=1,\dots,L} J(\mathbf{v}_i)$

---

objective is to find the values of $\boldsymbol{\rho}$ for which the objective function in (13) is minimum, which was solved using PSO. The dimension of each particle is equal to n. As $\boldsymbol{\rho}$ follows the axioms of the probability as discussed in Section 3.2, the same procedure is followed for updating the particle position in PSO algorithm. Once $\boldsymbol{\rho}$ is found, (12) is used to select the corresponding decision rule for detecting the hypothesis. The algorithmic frame work of PSO for finding $\boldsymbol{\rho}$ is presented in Algorithm 2, where the complexity of the Algorithm 2 is in the order of $O(nLK)$. where n is the number of the selected decision rules, L is the number particles in the swarm and K is the number of iterations that the PSO algorithm is repeated.

4. Experimental study

The objective of this section is twofold: first, demonstration of the proposed technique for improving the classification performance of randomly generated 3-class sample data is presented. Second, the performance of the proposed technique in the classification of a real data is discussed. All the experiments are performed on an Intel(R) Xeon(R) CPU with 3.5 GHz clock speed. The software package used is MATLAB 2018b. The parameters of PSO, while solving for $\hat{\boldsymbol{\pi}}$, are fixed as follows: the swarm size $L = 100$, the number of iterations $K = 100$, the individual learning $c_1 = 0.5$ and the group learning rate $c_2 = 1$. The parameters of the PSO algorithm, while solving for $\boldsymbol{\rho}$, are set as follows: $L = 200$, Number of iterations $K = 100$, $c_1 = 0.5$ and $c_2 = 1$.

4.1. Monte-Carlo simulation on solving mini-max technique using PSO with synthetic data

The data of 50 000 samples were generated as the outcomes of the discrete random variable X which takes the values $\{X = x | x \in \{1, 2, 3\}\}$ using some random prior probabilities. These prior probabilities were generated randomly with uniform distribution and normalized to make their sum equal to one. The data were assumed to be sent through a channel (classifier) whose characteristics are considered to be known to the observer. The channel matrix was generated randomly such that the ith element in the jth column is equal to the transition probability $P(j|i)$. Using this channel matrix, the outcomes of the discrete random variable Y, where $\{Y = y | y \in \{1, 2, 3\}\}$, were generated based on the roulette wheel selection process. When the roulette wheel is rotated, the pointer will stop in the region r_i with a probability of P_i. The roulette wheel selection process simulated to generate the output sample at the receiver y is as follows: Consider the input data sample x is equal to i, $i = 1$, 2, and 3. A random number r was generated which is uniformly distributed in $(0, 1)$. Then the output observation y is given as

$$y = \begin{cases} 1, & \text{if } 0 < r \le P(1|i) \\ 2, & \text{if } P(1|i) < r \le P(1|i) + P(2|i) \\ 3, & \text{if } P(1|i) + P(2|i) \le r < 1 \end{cases}$$

PSO was used to find the mini-max prior probabilities based on the channel matrix as explained in 3.2. Fig. 5 demonstrates the convergence of the PSO algorithm for an arbitrary channel matrix. The movement of the particles to the optimum solution $\hat{\boldsymbol{\pi}}$ can be seen in Fig. 5(a)–(d) and it can be observed that all the particles converge to a single point over the iterations. Convergence of the functional value $\frac{1}{J(\hat{\boldsymbol{\pi}})}$ to the maximum of the Bayes cost $V(\hat{\boldsymbol{\pi}})$ can be seen in Fig. 5(e)–(h).

Two different scenarios were considered to demonstrate the effectiveness of the usage of Mini-max technique on the probability of detection. In the first scenario, the data were assumed to be transmitted through a channel with different prior probabilities in each attempt. Since the mini-max technique depends only on the channel matrix and not on the prior probabilities, the change in the probability of detection will be decidedly less with the change in prior probabilities. The Mini-max technique is applicable only when the channel is noisy, and the probability of detection of the channel is less. In order to illustrate this, a randomly generated ternary-channel matrix was chosen, and the prior probabilities were generated randomly with uniform distribution. 100 attempts were made and in each attempt the prior probabilities were changed randomly and the channel matrix is fixed. It can be seen in Fig. 6(a) that the Mini-max technique has very less variance in the probability of Correct decision and it is improved in all of the attempts. It can be observed from Fig. 6(a) that the probability of Correct decision is having more fluctuations without using the mini-max technique. In the second scenario, both the channel matrix and the prior probabilities are randomly generated. Even in this case also, if the channel is noisy channel then the Mini-max technique will improve the probability of Correct decision. To illustrate this, 100 attempts were made with different channel matrices and prior probabilities in each attempt. It can be seen from Fig. 6(b) that the probability of Correct decision is improved in 80% of the attempts by using the Mini-max technique based randomized decision rule.

4.2. Experimental study on real-world datasets

4.2.1. Datasets

The proposed technique was used to improve the performance of the classifiers in real world classification problems. In order to test the performance of the proposed technique on real world problems, the datasets such as Abalone, Contraceptive methods choice, Teaching evaluation and Drug Consumption, which are collected from UCI database (Dua & Karra Taniskidou, 2017), Machinery Fault Dataset (Ribeiro, 2016), Dry Bean Dataset (Koklu & Ozkan, 2020) are used in the experiments. Detailed description of these datasets are given in Table 1. The data is divided into train and test sets such that the train set is balanced among all the classes. The classifier is trained on the train set and the channel matrix is formed based on the true labels and the predicted labels of the train set as shown in Fig. 2. Using this channel matrix, the Maximum Bayes Cost and the associated decision rules are obtained using the proposed technique. The labels of the test data are found using the associated decision rules with the corresponding probabilities. The sensor values of the Machinery Fault dataset are normalized to the standard normal distribution.

The proposed technique is applied on the prediction results of different classification techniques such as ANN, SVM (Allwein et al., 2000), NB (Hastie et al., 2009) and Ensemble of Trees (ENS) (Hastie et al., 2009) for all the datasets.

4.2.2. Performance evaluation

The performance of the classifier, in general, is evaluated using the classification accuracy. The classification accuracy of a classifier is defined as the percentage of the total number of correctly classified samples out the total available samples. In case of the binary classification the classification accuracy can be defined using the True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FP) as $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100$, where as in case of multi-class classification, the classification accuracy can be defined as given in (14).

$$Accuracy = \frac{TP_1 + TP_2 + \cdots + TP_M}{TP_1 + TP_2 + \cdots + TP_M + FP_1 + FP_2 + \cdots + FP_M} \times 100 \quad (14)$$

The classification accuracy may not always be an effective metric and misleading in case of the imbalanced data. Hence, in order prove the efficiency of the proposed Mini-max technique, several other classification metrics are used in this regard such as Macro average (Precision, Recall and F-score), Weighted average (Precision, Recall and F-score) and Cohen's Kappa score. Precision gives the positive predicted value, Recall shows the true positive rate and the F-score implies the harmonic average of the Precision and Recall (Hallajian et al., 2022). The Macro average Precision, Recall and F-score are computed by (15)–(17).

$$MAP = \frac{Prec_1 + Prec_2 + \cdots + Prec_M}{M} \quad (15)$$

$$MAR = \frac{Recall_1 + Recall_2 + \cdots + Recall_M}{M} \quad (16)$$

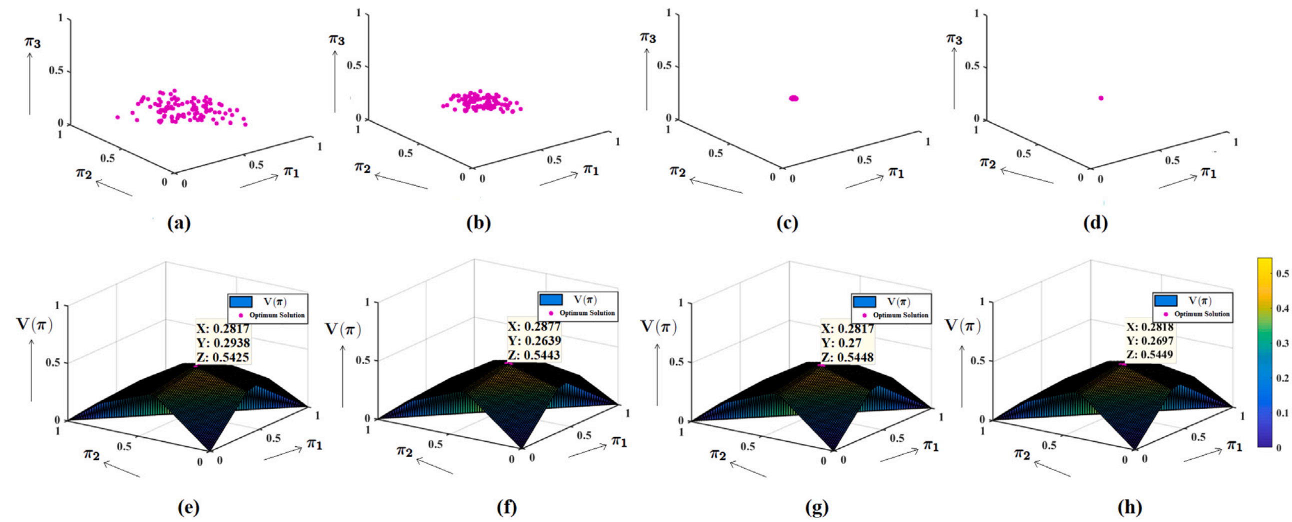

Fig. 5. Illustration of the convergence of PSO. (a)–(d) Particle movement with increase in iteration ((a) after 25 iterations, (b) 50 iterations, (c) 75 iterations and (d) 100 iterations) and (e)–(f)Functional value movement with increase in iteration ((e) after 25 iterations, (f) 50 iterations, (g) 75 iterations and (h) 100 iterations). It can be observed that he value of $V(\boldsymbol{\pi})$ has increased with increase in iteration ((e) 0.5425, (f) 0.5443, (g) 0.5448 and (f) 0.5449).

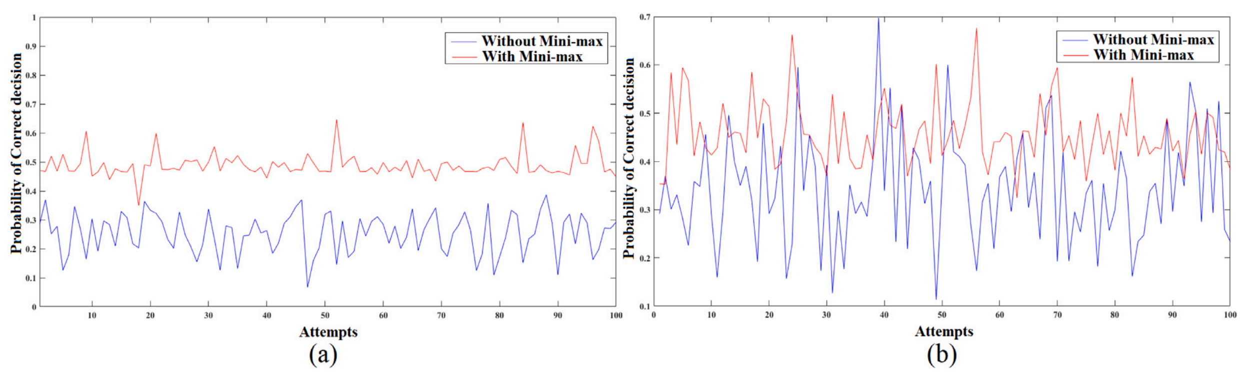

Fig. 6. Illustration of the improvement in the detection rate with the proposed mini-max technique for 100 attempts over (a) Fixed channel matrix with different prior probabilities (b) Distinct channel matrices with different prior probabilities.

Table 1
Description of the datasets (# Attributes - Number of Attributes, # Classes - Number of Classes and # Samples - Total Number of Samples in each dataset.)

Dataset	# Attributes	Attribute characteristics	# Samples	# Classes	Reference
Abalone	8	Categorical, Integer, Real	4177	3	Dua and Karra Taniskidou (2017)
Contraceptive method choice	9	Categorical, Integer	1473	3	Dua and Karra Taniskidou (2017)
Teaching assistant evaluation	5	Categorical, Integer	151	3	Dua and Karra Taniskidou (2017)
Drug consumption	12	Real	1885	7	Dua and Karra Taniskidou (2017)
Machinery fault	9	Real	1909	6	Ribeiro (2016)
Dry bean	16	Real	13 611	7	Koklu and Ozkan (2020)

$$MAF = 2 \times \frac{MAP \times MAR}{MAP + MAR} \tag{17}$$

where, $Prec_n$ and $Recall_n$ are the precision and recall of the class $n, \forall n = 1$ to M, MAP is the Weighted Average Precision, MAR is the Weighted Average Recall and MAF is the Weighted Average F-score. Similarly, the Weighted average Precision, Recall and F-score are computed by (18)–(20).

$$WAP = \frac{Prec_1 \times N_1 + Prec_2 \times N_2 + \cdots + Prec_M \times N_M}{\text{Total Number of Samples}} \tag{18}$$

$$WAR = \frac{Recall_1 \times N_1 + Recall_2 \times N_2 + \cdots + Recall_M \times N_M}{\text{Total Number of Samples}} \tag{19}$$

$$WAF = 2 \times \frac{WAP \times WAR}{WAP + MAR} \tag{20}$$

where WAP is the Weighted Average Precision, WAR is the Weighted Average Recall, WAF is the Weighted Average F-score and N_n is the number of samples in class n. Cohen's Kappa score (κ) (Landis & Koch, 1977) is a statistic that is used to measure inter-class reliability (and also intra-class reliability) for qualitative (categorical) items. It can handle both multi-class and imbalanced problems. The Cohen's Kappa score is defined by (21).

$$\kappa = \frac{P_0 - P_e}{1 - P_e} \tag{21}$$

where P_0 is an observational probability of agreement between actual and predicted values and P_e is a hypothetical expected probability of agreement under total independence of observer classifications.

4.2.3. Experimental results and analysis

To study the effectiveness of the proposed Mini-max technique, the experiments are performed for 50 times with different train and test combinations of the data. The box-plots of the classification accuracy

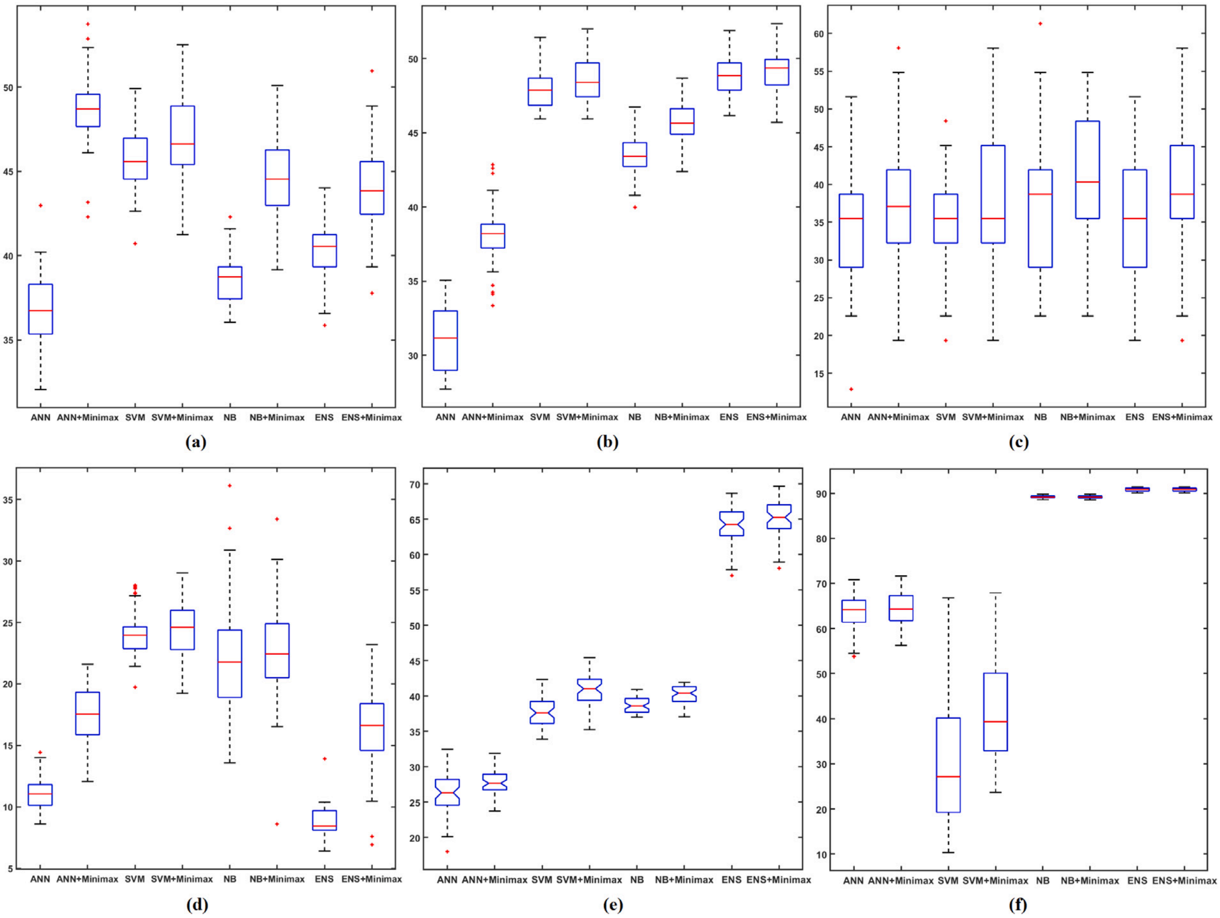

Fig. 7. Distribution of Classification Accuracy using the proposed mini-max technique applied to different classification techniques on different datasets for 50 attempts. (a) Abalone dataset (b) Contraceptive Method choice dataset (c) Teaching Assistant Evaluation dataset (d) Drug Consumption dataset (e) Machinery Fault dataset and (f) Dry Bean dataset.

Table 2
Comparison of the accuracy of different classifiers with mini-max technique.

Dataset	ANN	ANN + Mini-max	SVM	SVM + Mini-max	NB	NB + Mini-max	ENS	ENS + Mini-max
Abalone	36.79(2.12)	**48.70**(2.08)	45.67(1.84)	**47.06**(2.51)	38.60(1.44)	**44.67**(2.57)	40.31(1.60)	**43.90**(2.50)
Contraceptive method choice	31.13(2.13)	**38.11**(1.97)	47.95(1.33)	**48.50**(1.42)	43.41(1.32)	**45.65**(1.39)	48.77(1.27)	**49.16**(1.28)
Teaching assistant evaluation	34.65(7.71)	**37.81**(9.24)	35.35(6.48)	**38.19**(9.41)	37.81(8.14)	**41.03**(8.4)	35.48(7.40)	**39.10**(7.76)
Drug consumption	11.15(1.32)	**17.45**(2.41)	24.14(1.90)	**24.55**(2.01)	21.99(4.38)	**22.37**(3.84)	8.76(1.27)	**16.33**(3.35)
Machinery fault	26.38(2.95)	**27.78**(1.83)	37.70(2.05)	**40.73**(2.38)	38.71(1.12)	**40.21**(1.26)	63.96(2.66)	63.96(2.66)
Dry bean	63.79(4.29)	**64.50**(4.02)	30.78(15.3)	**41.32**(11.46)	89.16(0.26)	89.16(0.26)	90.84(0.38)	90.84(0.38)

of the classifiers and the proposed technique applied on the corresponding classifier for all the 50 attempts are shown in Fig. 7. Where ANN+Minimax, SVM+Minimax, NB+Minimax and ENS+Minimax are the proposed technique applied on the results of the classifiers ANN, SVM, NB and ENS respectively. The mean value and the standard deviation of the results of all the attempts are presented in Table 2. From Fig. 7 and Table 2, it can be observed that the proposed technique is performing better than the state-of-the art classifiers for all the datasets used in the study, except for the Dry Bean dataset, where the NB and ENS techniques have already achieved a mean accuracy of 89.16% and 90.84%, respectively. It can be observed from Table 2, that the proposed technique has improved the classification accuracy by 10% in some instances.

Fig. 8 shows different statistics of the predicted results of one of the attempts with and without mini-max technique. It is evident from Fig. 8 that the proposed methodology improves the performance of the classifier which is not performing well. When the classifier is performing well, the partition set $\Gamma_i = \{i\}$ will have the highest probability of getting selected as the Bayes rule. Hence the proposed technique is not effective when the classifier itself is performing well. This can be seen in Fig. 8 where, the NB and ENS classifiers has already achieved good performance for Dry Bean dataset, hence there is little improvement by the proposed technique.

The Macro average (Precision, Recall and F-score), Weighted average (Precision, Recall, F-score) and the Cohen's Kappa score (κ) of the classifiers and the proposed technique are presented in Table 3. The F-score of both Macro average and Weighted average of the proposed technique is improved when compared to that of the respective classifier, even though the Precision or Recall are not improved with the proposed technique at some instances. The macro and weighted average F-scores of the proposed technique for the Teaching Assistant Evaluation Dataset are improved by more than 10% even though the classification accuracy is improved by 3%–4%. The Cohen's Kappa score (κ) ranges from $-\frac{P_e}{1-P_e}$ to 1. The closer the value of κ to 1, the better the performance of the classifier. From Table 3, it can be observed that the proposed technique has produced an improved value of κ compared to that of the respective classifier.

Even though the mean performance of the proposed method is improving, the standard deviation is large when compared to that of the corresponding classifier. Hence, the statistical analysis is performed and

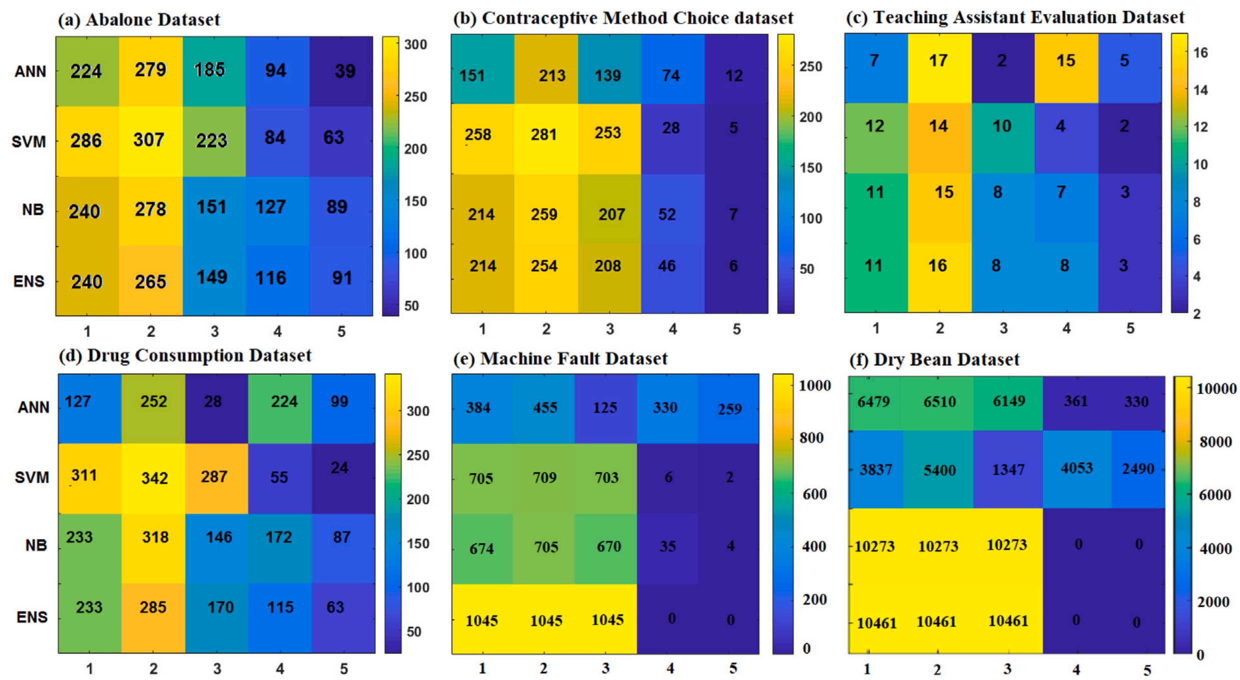

Fig. 8. Statistical representation of the predicted data for different datasets where the columns in each image represent (1) number of samples correctly predicted without Mini-max, (2) number of samples correctly predicted with Mini-max, (3) number of common samples correctly predicted with and without Mini-max, (4) number of samples mis-classified without applying Mini-max and correctly classified with Mini-max, and (5) number of samples correctly classified without Mini-max and mis-classified with Mini-max and each row represents the classifier used.

the z-test results are presented in Table 4. The z-test is presented instead of the t-test, since the t-distribution becomes normal distribution if the degrees of freedom is more than 30. The p-value denotes the level of the statistical significance of the proposed technique. The results of the proposed technique are considered as significant if the p-value is less than 0.05. The p-values of the proposed technique, presented in Table 4, are less than 0.05 in most of the cases. Hence, it can be concluded that the results obtained by the proposed technique are statistically significant for all the datasets except for the Drug Consumption dataset and Dry Bean dataset, since the p-values of the proposed technique are not less than 0.05 for these datasets when applied on the results of some classifiers.

4.3. Experimental results with human decision system on predicting cricket sport event

In order to validate the performance of the proposed method, it was used in human decision system, where soft decisions have to be taken, in order to improve the human capability of predicting the outcome. The prediction of the possible outcome in a cricket sport is presented in this section as an example. The possible outcomes considered are 0 (dot ball), $1, 2, 3, 4, 6$ or $wicket$. These outcomes are categorized into four classes, they are $class1 = \{0\}$, $class2 = \{1, 2, 3\}$, $class3 = \{4, 6\}$ and $class4 = \{wicket\}$. The human being is considered as the classifier where he has to predict the outcome of the next ball belongs to which one of the above four classes. Two different formats of the cricket game have been considered for prediction, one is 20 over match (T20I) and the other one is a 50 over match (ODI) and each over has 6 balls to be bowled. The prediction accuracy of the human being can be obtained by comparing the predicted outcomes with the actual outcomes.

The human prediction will improve based on the experience. In the similar way, the mini-max method can also be trained based on the experience. i.e., the channel matrix of the human prediction is initialized with enough amount of the data in order to have the enough information about the human prediction ability and there after it is modified at the end of every over based on the prediction by the human. The new decision rules can be obtained based on the modified channel matrix and the outcomes can be obtained based on these decision rules for the next over. In order to do this, the human being has to track his predictions and the actual outcomes of every over.

Experiments have been performed on the data from two different matches, a T20I match between Srilanka and South Africa and a ODI match between India and Bangladesh. The channel matrix is initialized with 80% of the data for both the matches. Fig. 9(a) and (b) shows the original labels (first row), the predicted outcomes by the human (Second row) and mini-max techniques (third row) for T20I and ODI respectively. The prediction accuracy of the human and mini-max techniques for the T20I are 26.83% and 46.34%, respectively and for the ODI are 41.29% and 46.55%, respectively.

The experiments are repeated for 50 times, in order to ensure the consistency of the mini-max technique. The prediction accuracy of the mini-max technique for the 20 over match, has a mean of 29.17% (with standard deviation of 5.57) which is better than the human prediction. Similarly, the mean of the prediction accuracy for the 50 over match is 41.29 (with standard deviation of 2.44). This shows that the proposed method is improving the prediction accuracy for the 20 over match whereas it failed to improve for the 50 over match. This is because of the fact that the 50 over match data is having the significantly less amount of the samples of $class3$ and $class4$ when compared to the $class1$ and $class2$, whereas in the 20 over match data, it is not the case.

5. Conclusion

The classification accuracy of the constructed DMC using the confusion matrix of the multi-class classifier has been increased using the proposed technique for real and the synthetic data set. The proposed technique is able to improve the classification accuracy by about 5% in most cases and more than 10% in some cases. The performance of the proposed technique is improved in terms of the metrics like

Table 3
Performance comparison of the proposed technique over the state-of-art classifiers in one of the attempts in terms of Macro and Weighted Average (Precision, Recall and F-Score) and Cohen's Kappa Score (κ). The bold text indicate the improved F-score and the kappa score by applying the proposed technique on the classifier's result.

Dataset	Classifier	Macro average			Weighted average			κ
		MAP	MAR	MAF	WAP	WAR	WAF	
Abalone	ANN	0.60	0.51	00.42	0.69	0.39	0.39	0.19
	ANN + Minimax	0.54	0.48	**0.48**	0.60	0.48	**0.52**	**0.20**
	SVM	0.56	0.62	0.50	0.64	0.50	0.47	0.25
	SVM + Minimax	0.53	0.55	**0.52**	0.58	0.53	**0.54**	**0.30**
	NB	0.51	0.57	0.42	0.58	0.42	0.35	0.16
	NB + Minimax	0.47	0.47	**0.46**	0.52	0.48	**0.50**	**0.23**
	ENS	0.51	0.57	0.42	0.58	0.42	0.35	0.13
	ENS + Minimax	0.45	0.46	**0.44**	0.49	0.46	**0.47**	**0.23**
Contraceptive method choice	ANN	0.49	0.41	0.27	0.69	0.26	0.34	0.09
	ANN + Minimax	0.40	0.39	**0.33**	0.57	0.37	**0.43**	0.09
	SVM	0.45	0.50	0.40	0.63	0.45	0.49	0.17
	SVM + Minimax	0.45	0.52	**0.43**	0.61	0.49	**0.53**	**0.20**
	NB	0.43	0.47	0.35	0.60	0.37	0.42	0.12
	NB + Minimax	0.41	0.46	**0.39**	0.56	0.45	**0.49**	**0.13**
	ENS	0.43	0.47	0.35	0.60	0.37	0.42	0.12
	ENS + Minimax	0.42	0.46	**0.39**	0.56	0.44	**0.48**	**0.13**
Teaching assistant evaluation	ANN	0.30	0.22	0.24	0.31	0.23	0.25	−0.15
	ANN + Minimax	0.57	0.57	**0.55**	0.57	0.55	**0.54**	**0.33**
	SVM	0.30	0.43	0.32	0.28	0.39	0.29	0.13
	SVM + Minimax	0.48	0.47	**0.44**	0.47	0.45	**0.43**	**0.18**
	NB	0.31	0.40	0.29	0.29	0.35	0.27	0.08
	NB + Minimax	0.55	0.49	**0.50**	0.55	0.48	**0.50**	**0.23**
	ENS	0.31	0.40	0.29	0.29	0.35	0.27	0.08
	ENS + Minimax	0.58	0.52	**0.52**	0.59	0.52	**0.52**	**0.28**
Drug consumption	ANN	0.18	0.18	0.10	0.33	0.10	0.10	0.02
	ANN + Minimax	0.18	0.17	**0.15**	0.33	0.20	**0.23**	**0.04**
	SVM	0.25	0.26	0.21	0.46	0.25	0.28	0.13
	SVM + Minimax	0.25	0.26	**0.22**	0.43	0.27	**0.31**	**0.14**
	NB	0.24	0.25	0.16	0.44	0.19	0.18	0.09
	NB + Minimax	0.22	0.22	**0.19**	0.41	0.25	**0.29**	**0.11**
	ENS	0.24	0.25	0.16	0.44	0.19	0.18	0.09
	ENS + Minimax	0.21	0.23	**0.19**	0.39	0.23	**0.26**	**0.10**
Machinery fault dataset	ANN	0.22	0.20	0.18	0.30	0.22	0.21	0.06
	ANN + Minimax	0.25	0.23	**0.24**	0.34	0.30	**0.31**	**0.12**
	SVM	0.39	0.44	0.34	0.48	0.34	0.37	0.24
	SVM + Minimax	0.39	0.38	**0.35**	0.48	0.37	**0.40**	**0.25**
	NB	0.43	0.39	0.35	0.57	0.37	0.42	0.27
	NB + Minimax	0.41	0.41	**0.36**	0.52	0.40	**0.45**	**0.28**
	ENS	0.62	0.69	0.61	0.76	0.68	0.70	0.60
	ENS + Minimax	0.62	0.69	0.61	0.76	0.68	0.70	0.60
Dry bean	ANN	0.57	0.61	0.54	0.67	0.56	0.58	0.48
	ANN + Minimax	0.57	0.61	0.54	0.66	0.57	0.58	0.48
	SVM	0.18	0.20	0.18	0.27	0.33	0.29	0.18
	SVM + Minimax	0.47	0.51	**0.42**	0.56	0.47	**0.48**	**0.37**
	NB	0.90	0.91	0.90	0.89	0.89	0.89	0.87
	NB + Minimax	0.90	0.91	0.90	0.89	0.89	0.89	0.87
	ENS	0.92	0.92	0.92	0.91	0.91	0.91	0.89
	ENS + Minimax	0.92	0.92	0.92	0.91	0.91	0.91	0.89

Table 4
Z-test results of the Mini-max technique when applied to the corresponding classifier.

Dataset	ANN + Mini-max vs. ANN		SVM + Mini-max vs. SVM		NB + Mini-max vs. NB		ENS + Mini-max vs. ENS	
	z-value	p-value	z-value	p-value	z-value	p-value	z-value	p-value
Abalone	28.367	<0.0001	3.1458	<0.0001	14.5577	<0.0001	8.536	<0.0001
Contraceptive method choice	16.996	<0.0001	1.9847	0.0236	8.0101	<0.0001	1.5292	0.0631
Teaching assistant evaluation	1.8579	0.0316	1.7569	0.0395	1.95	0.0256	2.3828	0.0086
Drug consumption	16.2074	<0.0001	1.0382	0.1496	0.4632	0.3216	14.923	<0.0001
Machinery fault dataset	2.8611	0.0021	6.8169	<0.0001	6.3157	<0.0001	1.8853	0.0297
Dry bean	0.8556	0.1961	3.8992	<0.0001	−0.3201	0.6256	−0.0522	0.5208

Precision, Recall, F-score and Cohen's Kappa score. This paves the way to use the proposed technique to improve the multi-class classification accuracy. The proposed technique can be viewed as the analogy of the improvement of the human decision based on the earlier decisions taken. More exposure to real time decisions lead to new channel matrix of the DMC and hence modified randomized rule constructed using the proposed technique can be used to improve the overall probability of Correct decision, which is identified as the continuous learning. The theoretical framework of the extension of the proposed model as the continuous learning is left as the natural extension of the proposed technique. The proposed technique can be used as the digital aid to support day-to-day decision making activities of the human.

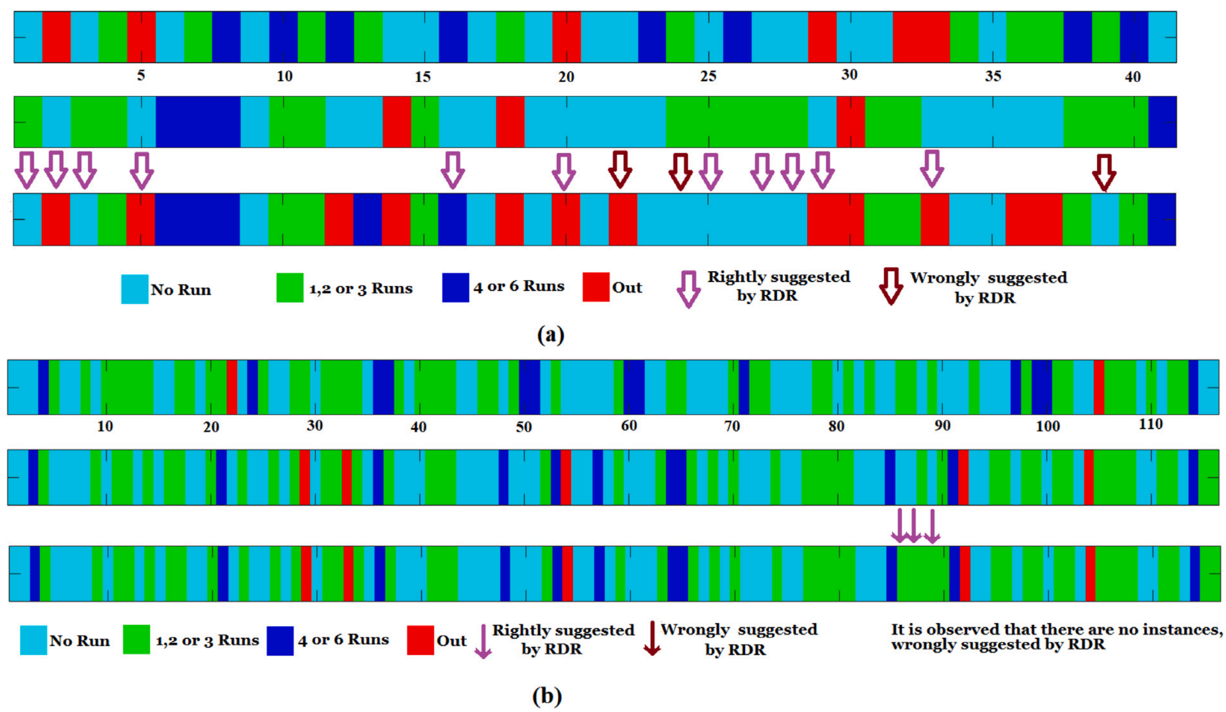

Fig. 9. Illustration of the better suggestions predicted by the proposed mini-max technique for the (a) 20 over match and (b) 50 over match. In both the figures, the first row represent the actual event happened, the second row represent the event predicted by the human and the third row represent the event suggested by the proposed mini-max technique.

Table A.5
Number of distinct partitions of the observation set Γ for $M = 2, 3, 4, \& 5$.

M = 2:			M = 3:
Number of elements in each partition	Permutations	Number of elements in each partition	Permutations
1, 1	${}^2P_2 = 2$	1,1,1	${}^3P_3 = 6$
2, 0	${}^2P_1 = 2$	2,1,0	${}^3C_2 \times {}^3P_2 = 18$
		3,0,0	${}^3P_1 = 3$
$Total = 4 = 2^2$		$Total = 27 = 3^3$	
M = 4:			M = 5:
Number of elements in each partition	Permutations	Number of elements in each partition	Permutations
1,1,1,1	${}^4P_4 = 24$	1,1,1,1,1	${}^5P_5 = 120$
2,1,1,0	${}^4C_2 \times {}^4P_3 = 144$	2,1,1,1,0	${}^5C_2 \times {}^5P_4 = 1200$
2,2,0,0	$(\frac{1}{2}) \times {}^4C_2 \times {}^4P_2 = 36$	2,2,1,0,0	$(\frac{1}{2}) \times {}^5C_2 \times {}^3C_2 \times {}^5P_3 = 900$
3,1,0,0	${}^4C_3 \times {}^4P_2 = 48$	3,1,1,0,0	${}^5C_3 \times {}^5P_3 = 600$
4,0,0,0	${}^4P_1 = 4$	3,2,0,0,0	${}^5C_3 \times {}^5P_2 = 200$
		4,1,0,0,0	$5 \times {}^5P_2 = 100$
		5,0,0,0,0	${}^5P_1 = 5$
$Total = 256 = 4^4$		$Total = 3125 = 5^5$	

CRediT authorship contribution statement

Rajasekharreddy Poreddy: Methodology, Software, Investigation, Resources, Data curation, Writing – original draft. **E.S. Gopi:** Conceptualization, Conceptualization, Writing – review, Visualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix. Decision rules for M hypotheses

It is mentioned in Section 2.2 that the number of decision rules for M hypotheses case are M^M. Here, it is proved deductively for different values of M based on the partitions of the observation set as presented in Table A.5.

Where ${}^nP_r$ represents the number of ordered sets, r from a group of n sets. ${}^nC_r$ represents the selection of r sets (hypotheses or classes) from a group of n sets.

References

Ahuja, B., & Vishwakarma, V. P. (2021). Deterministic multi-kernel based extreme learning machine for pattern classification. *IEEE Transactions on Cybernetics*, *183*, 1–12. http://dx.doi.org/10.1016/j.eswa.2021.115308.

Allwein, E., Schapire, R., & Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, *1*, 113–141.

Barbosa, H. J. C. (2012). Decision-making in recommender systems: The role of user's goals and bounded resources. In *Workshop on human decision making in recommender systems in 6th acm conference on recommender systems* (pp. 1–7).

Chai, Z., Li, Y., & Zhu, S. (2021). P-MOIA-RS: a multi-objective optimization and decision-making algorithm for recommendation systems. *Journal of Ambient Intelligence and Humanized Computing*, *12*(1), 443–454. http://dx.doi.org/10.1007/s12652-020-01997-x.

Chale, M., & Bastian, N. D. (2022). Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems. *Expert Systems with Applications*, *207*, 1–18. http://dx.doi.org/10.1016/j.eswa.2022.117936.

Chen, L., Gemmis, M. D., Felfernig, A., Lops, P., Francesco, R., & Semeraro, G. (2013). Human decision making and recommender systems. *ACM Transactions on Interactive Intelligent Systems*, *3*(3), 17:1–17:7.

Dua, D., & Karra Taniskidou, E. (2017). UCI machine learning repository, science. Available at: http://archive.ics.uci.edu/ml.

Felipe, D. P., Rodrigo, T. N. C., Gustavo, P. H., & Wendel, M. D. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, *115*, 635–655. http://dx.doi.org/10.1016/j.eswa.2018.08.003.

Gopi, E. S. (2007). *Algorithm collections for digital signal processing applications using matlab* (pp. 1–8). Netherlands: Springer.

Gopi, E. S. (2016). *Digital signal processing for wireless communication using matlab* (pp. 51–86). Switzerland: Springer.

Hallajian, B., Motameni, H., & Akbari, E. (2022). Ensemble feature selection using distance-based supervised and unsupervised methods in binary classification. *Expert Systems with Applications*, *200*, 1–18. http://dx.doi.org/10.1016/j.eswa.2022.116794.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (pp. 605–624). New York: Springer-Verlag.

Haung, G., Song, S., Gupta, J., & Cheng, W. (2014). Semi-supervised and unsupervised extreme learning machines. *IEEE Transactions on Cybernetics*, *44*(12), 2405–2417. http://dx.doi.org/10.1109/TCYB.2014.2307349.

Irfan, M., Jiangbin, Z., Ali, S., Iqbal, M., Masood, Z., & Hamid, U. (2021). DeepShip: An underwater acoustic benchmark dataset and a separable convolution based autoencoder for classification. *Expert Systems with Applications*, *183*, 1–12. http://dx.doi.org/10.1016/j.eswa.2021.115270.

Karla, O., Mello, M., Botelho, G., Deng, Z., Koushanfar, F., & Kiperstok, A. (2021). Water end-use consumption in low-income households: Evaluation of the impact of preprocessing on the construction of a classification model. *Expert Systems with Applications*, *185*, 1–15. http://dx.doi.org/10.1016/j.eswa.2021.115623.

Koklu, M., & Ozkan, I. A. (2020). Multiclass classification of dry beans using computer vision and machine learning techniques. *Computers and Electronics in Agriculture*, *174*, 1–11. http://dx.doi.org/10.1016/j.compag.2020.105507.

Krishna, N., Bhattu, S. N., Somayajulu, D., Kumar, N., & Reddy, K. (2022). GssMILP for anomaly classification in surveillance videos. *Expert Systems with Applications*, *203*, 1–14. http://dx.doi.org/10.1016/j.eswa.2022.117451.

Kudryavtsev, A., & Pavlodsky, J. (2012). Description-based and experience-based decisions: individual analysis. *Judgment and Decision Making*, *7*(3), 316–331.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174. http://dx.doi.org/10.2307/2529310.

Leberia, N., Blanco, A., Perez, A., & Casillas, A. (2022). Preliminary exploration of topic modelling representations for electronic health records coding according to the international classification of diseases in spanish. *Expert Systems with Applications*, *204*, 1–8. http://dx.doi.org/10.1016/j.eswa.2022.117303.

Marco, O., Nicola, U., Claudio, C., Silvia, B., & Giuseppe, D. (2022). On technical trading and social media indicators for cryptocurrency price classification through deep learning. *Expert Systems with Applications*, *198*, 1–15. http://dx.doi.org/10.1016/j.eswa.2022.116804.

Mostafa, K., & Mozaffar, R. (2022). Classification of seven Iranian wheat varieties using texture features. *Expert Systems with Applications*, *199*, 1–12. http://dx.doi.org/10.1016/j.eswa.2022.117014.

Muhammad, N., Nadeem, A., Alhudhaif, A., & Abdulrahman, A. (2021). Guaranteeing correctness of machine learning based decision making at higher educational institutions. *IEEE Access*, *9*, 92864–92880. http://dx.doi.org/10.1109/ACCESS.2021.3088901.

Newell, B., & Rakow, T. (2007). The role of experience in decisions from description. *Psychonomic Bulletin & Review*, *14*, 1133–1139.

Papoulis, A., & Pillai, S. U. (2002). *Probability, random variables and stochastic processes* (pp. 15–45). McGraw-Hill, Europe.

Poor, H. V. (1994). *An introduction to signal detection and estimation* (pp. 13–22). Springer-Verlog.

Ribeiro, F. M. L. (2016). Mafaulda - machinery fault database. Available at: http://www02.smt.ufrj.br/~offshore/mfs/page_01.html.

Rodriguez, V., Sanchez, M., & Chica, M. (2015). Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, *71*, 804–818. http://dx.doi.org/10.1016/j.oregeorev.2015.01.001.

Shamout, F., Zhu, T., & Clifton, D. A. (2021). Machine learning for clinical outcome prediction. *IEEE Reviews in Biomedical Engineering*, *14*, 116–126. http://dx.doi.org/10.1109/RBME.2020.3007816.

Singh, Y., & Biswas, A. (2022). Robustness of musical features on deep learning models for music genre classification. *Expert Systems with Applications*, *199*, 1–14. http://dx.doi.org/10.1016/j.eswa.2022.116879.

Yu-Lin, H., Gui-Liang, O., Phillippe, F., Huang, J., & Suganthan, P. (2022). A novel dependency-oriented mixed-attribute data classification method. *Expert Systems with Applications*, *199*, 1–18. http://dx.doi.org/10.1016/j.eswa.2022.116782.

Yuri, N., Vicente, G. D., Carlos, M., Camilo, G. C., & Ruben, G. C. (2019). Usage of machine learning for strategic decision making at higher educational institutions. *IEEE Access*, *7*, 75007–75017. http://dx.doi.org/10.1109/ACCESS.2019.2919343.