An Online Learning Algorithm for a Neuro-Fuzzy Classifier with Mixed-Attribute Data

Thanh Tung Khuat , *Student Member, IEEE*, and Bogdan Gabrys , *Senior Member, IEEE*

Abstract—General fuzzy min-max neural network (GFMMNN) is one of the efficient neuro-fuzzy systems for data classification. However, one of the downsides of its original learning algorithms is the inability to handle and learn from the mixed-attribute data. While categorical features encoding methods can be used with the GFMMNN learning algorithms, they exhibit a lot of shortcomings. Other approaches proposed in the literature are not suitable for on-line learning as they require entire training data available in the learning phase. With the rapid change in the volume and velocity of streaming data in many application areas, it is increasingly required that the constructed models can learn and adapt to the continuous data changes in real-time without the need for their full retraining or access to the historical data. This paper proposes an extended online learning algorithm for the GFMMNN. The proposed method can handle the datasets with both continuous and categorical features. The extensive experiments confirmed superior and stable classification performance of the proposed approach in comparison to other relevant learning algorithms for the GFMM model.

Index Terms—General fuzzy min-max neural network, classification, mixed-attribute data, online learning, neuro-fuzzy classifier.

I. INTRODUCTION

CLASSICAL batch learning algorithms usually require the complete availability of data at the training time. These algorithms do not constantly accommodate new information to the built models. Instead, we need to reconstruct the model from scratch when the underlying data changes. This operation is time-consuming, especially in the case of massive data, and the constructed models are more likely to be outdated in dynamically changing environments. Taking an advertising recommendation system as an example, this system constructs a customer preference model based on the tracking information about the shopping and browsing behaviors of the users. The buying activities and preferences are temporary and continuously changing. For example, the pandemic such as COVID-19 has dramatically changed the online shopping behaviors of customers where people tend to purchase things they have never bought before. Therefore, the learning models trained on consumer behavior data prior to the pandemic have been deteriorated or crashed. As a result, these models need to be retrained on new (normal?) behavior data. In this context, and many others characterised by streaming data in changing environments, it is desirable or even necessary to have online learning algorithms that can learn constantly new information without retraining from scratch.

T.T. Khuat (email: thanhtung.khuat@student.uts.edu.au) and B. Gabrys (email: Bogdan.Gabrys@uts.edu.au) are with Advanced Analytics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia.

With the increase in the data volume and the rapid change of the environmental conditions nowadays, online learning algorithms are in high demand [1], [2]. These algorithms require smaller or no data storage as they only need one or few newest training samples at one time to rapidly update the constructed model. Hence, the online learning models are ideal candidates for the systems with frequently updating demands. General fuzzy min-max (GFMM) neural network [3], [4] is such an incremental learning model, which can be effectively utilized for data classification problems. This type of learning model combines the artificial neural network with the fuzzy set theory to form a consolidated framework. The model creates new hyperboxes or adjusts the existing hyperboxes to cover new samples in its structure. Each hyperbox is defined by the minimum and maximum points in an n-dimensional space. The degree-of-fit of an input pattern to a hyperbox is identified by a membership function.

GFMM neural network [3] is a significant enhancement of the FMNN [5]. Unlike the FMNN, the GFMM model can handle the uncertainty associated with the input data by accepting the input patterns not only as single points but also hyperboxes. In addition, it can handle both labeled and unlabeled data samples in a single model. The GFMMNN still maintains the online learning ability from the FMNN using a single-pass through training samples learning algorithms to expand or create new hyperboxes. To avoid the ambiguity in the classification phase, the original online learning algorithm proposed in [3] does not allow the overlap between hyperboxes representing different classes. Therefore, after expanding a hyperbox to cover the input pattern, a hyperbox contraction procedure must be performed if there is an overlapping region between two hyperboxes belonging to different classes. However, the hyperbox contraction operation can lead to undesirable classification errors as shown in the original paper and subsequent publications [3], [6], [7], [8], [9]. As a result, in a recent study, we have proposed an improved online learning algorithm for the GFMM model (IOL-GFMM) [9] to overcome this limitation by not using the hyperbox contraction step during the learning process. This algorithm integrates the strong points of the batch learning algorithm proposed in [6] and the incremental learning ability of the original algorithm into a single algorithm.

However, both the original online learning algorithm [3] and the IOL-GFMM algorithm [9] work well on the datasets with only numerical features. To perform classification for the datasets with mixed-type features, we would need to use the encoding methods to transform the categorical values into numerical values. As shown in a recent study [10],

each encoding method has its own drawbacks and except for the CatBoost [11] and label encoding techniques, all of the remaining encoding approaches need to use the entire training set to encode the categorical features. Therefore, they are not appropriate for incremental learning algorithms, where the new values can appear during the operation time. In addition, according to the empirical results in [10], the classification performance of the online learning algorithms using the CatBoost or label encoding method for the GFMM model is quite poor. It is because the label encoding method imposes an artificial distance metric for categorical groups, in which this distance is not correspondent to the correlation among original categorical values [12]. Not only this poses a serious problem but the CatBoost encoding method is sensitive to the order of training samples presentation and a shift in the encoded values between training and testing data as well as between training samples have been observed. For the same categorical value, its encoded value in the training data may be distinct from that in the testing data. Even in the training set, the same categorical value may be mapped into many different encoded values depending on the historical patterns prior to the current training pattern. Our proposed method in this paper avoids all of these issues by not using any encoding methods for discrete attributes in the first place.

Many real-world datasets are in the form of mixed-type features. The mixed-attribute data contain both continuous and discrete (or categorical) features. Nowadays, the mixed-attribute data are more and more popular in a wide range of applications from the credit approval data to medical diagnostic data [13]. Hence, to apply the GFMMNN to such problems, we need to extend its current learning algorithms so that they can deal effectively with mixed-attribute data. Although there are a large number of improved algorithms of the FMNN model, only two existing studies have focused on expanding the learning algorithms for both categorical and numerical features as shown in a recent survey paper [14]. The first study was proposed in [15] (denoted by Onln-GFMM-M1 in this paper) using the correlation between the occurrence frequency of categorical values and classes to determine the similarity degrees among categorical values for each categorical feature. After that, the authors proposed to extend the original online learning algorithm [3] for mixed-attribute data. The second idea of expanding the original online learning algorithm of the FMNN model for both numerical and categorical features was introduced in [16], called Onln-GFMM-M2 in this paper. It uses the one-hot encoding method for the categorical features and logical operators such as AND and OR to operate on the categorical groups. However, the main weak point of both algorithms is the use of the entire training set to encode or compute the similarity degree between categorical values. If a new value occurs without being encountered during a training process before, these algorithms cannot handle such situation and produce a valid prediction. Different from these two approaches, this paper proposes a new incremental learning algorithm for both continuous and categorical features. The proposed method does not use any encoding methods for categorical values. Instead, it uses a union operator of a set to add new categorical value to the current set of values in each

categorical feature of a hyperbox. The decision on expanding a selected hyperbox to accommodate a new input pattern is based on the change in the entropy for each categorical feature. We also modify the membership function to handle both categorical and numerical attributes. The membership degree for all categorical features is computed from the average probability of categorical values in the input sample with regard to all of the existing discrete values stored in discrete features of the hyperbox. In short, our main contribution in this paper can be summarized as follows:

- We propose a novel online learning algorithm for the GFMMNN able to learn from mixed-attribute data. To the best of our knowledge, this is the first online learning algorithm for the family of fuzzy min-max neural networks which can handle both continuous and categorical features without using any encoding methods.
- We present and prove several properties of the proposed method with regard to the categorical/discrete attributes.
- We conduct extensive experiments to prove the effectiveness of the proposed method in comparison to other relevant methods.
- We assess the impact of hyper-parameters on the classification performance of the proposed method and propose a simple method for the parameter estimation.

The rest of this paper is structured as follows. Section II summarizes briefly the architecture of the GFMMNN and its improved online learning algorithm. Section III is devoted to describing the proposed method and its properties. Experimental results and discussion are shown in Section IV. Section V concludes the key findings in this paper and informs potential research directions.

II. PRELIMINARIES

A. General fuzzy min-max neural network

The GFMMNN [3] are composed of three layers, i.e., input, hyperbox (hidden), and output layers. The input layer in the GFMM model can accept both real valued point and interval (hyperbox-typed) based input samples. If each input pattern has n dimensions, there will be $2n$ nodes in the input layer, in which the first n nodes are for the lower bounds and the remaining n nodes represent the upper bounds. The hidden layer contains hyperboxes dynamically generated in the learning process. The connection weights between the lower bound nodes and a hyperbox B_i form a vector V_i storing the minimum coordinates for that hyperbox. Similarly, the connection weights from the upper bound input nodes to a hyperbox B_i are represented by a vector W_i containing the maximum coordinates of that hyperbox. The values of matrices $\mathbf{V}$ and $\mathbf{W}$ for all hyperboxes are tuned during the learning process. Each hyperbox B_i in the hidden layer is fully connected to all output nodes. The connection weights between the hyperbox layer and output layer are kept in a matrix $\mathbf{U}$ and each of its element u_{ij} is computed as follows:

$$u_{ij} = \begin{cases} 1, \text{if } class(B_i) = c_j \\ 0, \text{otherwise} \end{cases} \quad (1)$$

where c_j is the j-th class node in the output layer.

Each hyperbox $B_i = [V_i, W_i]$, where $V_i = [v_{i1}, \ldots, v_{in}]$ and $W_i = [w_{i1}, \ldots, w_{in}]$ are the minimum and maximum points respectively, is associated with a membership function $b_i(X, B_i)$. This membership function is used to calculate the degree-of-fit for each input pattern $X = [X^l, X^u]$ to the hyperbox B_i, where $X^l = [x_1^l, \ldots, x_n^l]$ and $X^u = [x_1^u, \ldots, x_n^u]$ are the lower and upper bounds of an input pattern suitably normalised within an n-dimensional unit hypercube $[0, 1]^n$. The membership function is given as follows:

$$b_i(X, B_i) = \min_{j=1}^{n}(\min([1 - f(x_j^u - w_{ij}, \gamma_j)], \qquad (2)$$
$$[1 - f(v_{ij} - x_j^l, \gamma_j)]))$$

where $f(\xi, \gamma)$ is a ramp function defined in (3):

$$f(\xi, \gamma) = \begin{cases} 1, & \text{if } \xi \cdot \gamma > 1 \\ \xi \cdot \gamma, & \text{if } 0 \le \xi \cdot \gamma \le 1 \\ 0, & \text{if } \xi \cdot \gamma < 0 \end{cases} \qquad (3)$$

with $\gamma = (\gamma_1, \gamma_2, ..., \gamma_n)$ a sensitivity parameter controlling the decreasing speed of the membership degree, and $0 \le b_i \le 1$. If $b_i(X, B_i) = 1$, then X is fully contained in the core of the hyperbox B_i.

B. An improved online learning algorithm

The learning process in GFMM consists of creating and adjusting hyperbox fuzzy sets on the basis of the presented input patterns. There have been a number of fundamental GFMM learning algorithms proposed in the literature which fall into one of two key categories: (i) incremental/on-line learning algorithms where the hyperboxes are adjusted, if needed, after every presentation of a single pattern [3], [9] and (ii) batch learning algorithms which assume the full training data is available from the beginning of the training process for the training algorithm to use [6], [17], [7]. The performance of the original online algorithm [3] is sensitive to the order of training data presentation and the maximum hyperbox size hyper-parameter setting. When inappropriate maximum size of hyperbox is selected and combined with the existing hyperbox contraction process, it can lead to undesired classification errors as analysed and illustrated in [9]. Therefore, in a recent study, we proposed an improved version of the original online learning algorithm, in which, similarly to the agglomerative algorithms in [6], the contraction process is not used during the learning process. The algorithm contains only two main steps, i.e., creation or expansion of hyperboxes and overlap test. In the original online learning algorithm, if a selected hyperbox candidate fulfills the expansion condition related to the maximum hyperbox size, it will be expanded. Then, if the overlap between the newly expanded hyperbox and the existing hyperboxes belonging to other classes occurs, the relevant hyperboxes are contracted. In contrast, in the IOL-GFMM algorithm, if the undesired overlap would happen after the expansion, the selected candidate will not be expanded.

For a training sample $X = [X^l, X^u, c_X]$, the algorithm first filters all existing hyperboxes with the same class as c_X. After that, the membership values between X and these hyperboxes are computed and sorted in descending order.

Hyperbox candidates will be then checked with regard to meeting the expansion conditions beginning from the hyperbox with the highest membership degree. If the maximum membership value is one, i.e. X is contained in the hyperbox, the learning algorithm continues with the next training sample. Otherwise, the expansion condition checking process only terminates when there is a hyperbox candidate which can be expanded to cover X or no further hyperbox candidates exist. If none of the existing candidates can be expanded, a new hyperbox is generated with the same coordinates as X. The first expansion condition is the maximum hyperbox size. For the hyperbox candidate B_i, first, the maximum hyperbox size condition given in (4) is checked:

$$\max(w_{ij}, x_j^u) - \min(v_{ij}, x_j^l) \le \theta, \; \forall j \in [1, n] \qquad (4)$$

where n is the number of features. If this condition is met, the hyperbox B_i is temporarily expanded to new size as follows:

$$w_{ij}^{new} = \max(w_{ij}^{old}, x_j^u)$$
$$v_{ij}^{new} = \min(v_{ij}^{old}, x_j^l), \; \forall j \in [1, n] \qquad (5)$$

Then, the newly expanded B_i will be tested for undesired overlaps with all of the hyperboxes representing the other classes (i.e. different from the class associated with B_i). There are four overlap test cases shown in [3]. If there is no overlapping area occurring, the new size of B_i is kept. Otherwise, B_i is reverted to the coordinates before expanding and the next hyperbox candidate is considered.

For an unseen pattern, its predicted class is the class of the hyperbox representing the highest membership value for that input pattern among all existing hyperboxes in the model. In the case when many hyperboxes representing K different classes have the same maximum membership degree (b_{win}), an additional criterion is used to find the appropriate class for X. The final class of X is the class c_k with the highest score of $\mathcal{P}(c_k|X)$ given by:

$$\mathcal{P}(c_k|X) = \frac{\sum_{j \in \mathcal{I}_{win}^k} n_j \cdot b_j}{\sum_{i \in \mathcal{I}_{win}} n_i \cdot b_i} \qquad (6)$$

where $k \in [1, K]$ and $\mathcal{I}_{win} = \{i, \text{if } b_i = b_{win}\}$ comprises the indexes of all hyperboxes with the maximum membership value of b_{win}, $I_{win}^k = \{j, \text{if } class(B_j) = c_k \text{ and } b_j = b_{win}\}$ is a subset of I_{win} containing indexes of the k-th class, and n_i is the number of training samples covered by hyperbox B_i. We would like to refer the interested readers to references [9] and [18] for the detailed algorithm as well as its time complexity.

III. PROPOSED METHOD

A. Formal Description

Let $\mathcal{T}_N = \{(X_i^l, X_i^u, X_i^d, c_i)\}_{i=1}^{N}$ be N training patterns, where c_i is the class of the i-th pattern, $X_i^l = (x_{i1}^l, \ldots, x_{in}^l)$ and $X_i^u = (x_{i1}^u, \ldots, x_{in}^u)$ are n continuous attributes (determined in a unit hyper-cube $[0, 1]^n$) of lower bound X_i^l and upper bound X_i^u for the i-th training sample, $X_i^d = (x_{i1}^d, \ldots, x_{ir}^d)$ represent r discrete attributes for the i-th training sample, x_{ij}^d is a categorical value of the j-th categorical feature (A_j^d) at the i-th training sample, $x_{ij}^d \in \text{DOM}(A_j^d) = $

$\{a_{1j}, a_{2j}, \ldots, a_{n_j j}\}$, where $\text{DOM}(A_j^d)$ is a domain of discrete values for the categorical attribute A_j^d and n_j is the number of symbolic values of A_j^d. This paper proposes an online learning algorithm to train an efficient GFMM classifier from $\mathcal{T}_N$.
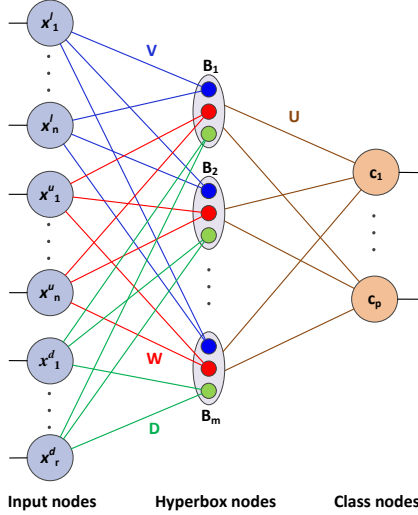

Fig. 1. The extended architecture of GFMMNN for mixed-attribute data

1) Architecture of GFMMNN for Mixed-Attribute Data:

First of all, we need to expand the architecture of the GFMMNN for mixed-attribute data. Instead of using $2n$ input nodes as in the GFMM model for continuous data, we will need $2n + r$ nodes for the input layer. The first $2n$ nodes are lower bound and upper bound nodes for n numerical features, respectively. The last r remaining nodes correspond to r categorical features in each input pattern. These r input nodes are connected to m hyperboxes by connection weights stored in a matrix $\mathbf{D}$. New architecture of the GFMMNN is shown in Fig. 1. Beside the minimum points V_i and the maximum points W_i, each hypebox B_i in the hidden layer also contains a vector D_i storing r discrete-valued sets. Each element $d_{ij} \in D_i$ $(1 \leq j \leq r)$ is a set of symbolic values with their cardinalities for the j-th categorical dimension of the hyperbox B_i. For example, $d_{i1} = \{apple : 5, orange : 1\}$ means that the first categorical feature of the hyperbox B_i contains 5 values of *apple* and 1 value of *orange*. The values of vectors V_i, W_i, and D_i for each hyperbox B_i are generated and adjusted during the learning process. The membership function of each hyperbox B_i with regard to each input pattern with mixed-attribute $X = (X^l, X^u, X^d, c_X)$ is modified as follows:

$$b_i(X, B_i) = \alpha \cdot \min_{j=1}^{n}(\min([1 - f(x_j^u - w_{ij}, \gamma_j)],$$
$$[1 - f(v_{ij} - x_j^l, \gamma_j)])) + \frac{1 - \alpha}{r} \cdot \sum_{j=1}^{r} \mathsf{P}_j(x_j^d \in d_{ij}) \quad (7)$$

where α $(\alpha \in [0,1])$ is a trade-off factor regulating the contribution level of numerical features part and categorical features part to the membership score, and $\mathsf{P}_j(x_j^d \in d_{ij})$ is a probability of encountering a symbolic value x_j^d in the

j-th categorical attribute of the hyperbox B_i. This probability is formally defined as follows:

$$\mathsf{P}_j(x_j^d \in d_{ij}) = \frac{|\{a \in d_{ij} | a = x_j^d\}|}{|d_{ij}|} \quad (8)$$

where $|\cdot|$ is the cardinality of a set. For the above example, we obtain $\mathsf{P}_j(\{orange\} \in d_{i1}) = 1/6$, $\mathsf{P}_j(\{apple\} \in d_{i1}) = 5/6$, and $\mathsf{P}_j(\{banana\} \in d_{i1}) = 0$. Unlike the numerical part in the membership function, we use an average operation for the categorical part to reduce sensitivity to the membership value. If we also use the $\min$ operator for the categorical part, the membership value for the categorical features will get the value of zero when there is only one discrete feature getting a new symbolic value.

2) Extended Improved Online Learning Algorithm for Mixed-Attribute Data Classification:

To create new hyperboxes or adjust existing hyperboxes towards learning mixed-attribute training samples in the GFMM model, we need to expand the current improved learning algorithm presented in subsection II-B, denoted by EIOL-GFMM in this paper. The proposed modifications include the expansion condition for categorical features, the way of accommodating a categorical value into the hyperbox, and the overlap test for categorical features.

For each training sample, $X = [X^l, X^u, X^d, c_X] \in \mathcal{T}_N$, the algorithm first filters all of the existing hyperboxes representing the same class as c_X. Then, the membership values of X in these selected hyperboxes are calculated and sorted in descending order. After that, in turn, we select expandable hyperbox candidates starting from the hyperbox with the highest membership degree if the highest membership score is smaller than one. Assuming that B_i is the currently considered hyperbox, the numerical features of B_i are checked for the maximum hyperbox size condition as shown in (4). If the expansion condition for continuous features is satisfied, the algorithm continues to check the constraint for discrete features.

Entropy-based measures can be used to assess the heterogeneity of data in clusters, and they are appropriate for clustering of categorical data due to the lack of explicit distance measures between discrete values [19]. We propose to use the change in the entropy value of categorical values contained in the hyperbox to decide whether the current hyperbox can be expanded to accommodate the categorical values of a new training sample. Given a categorical attribute j, let $H_j(B_i)$ be the current entropy of hyperbox B_i for the j-th categorical feature, computed from the probability of all current categorical values stored in the j-th attribute as follows:

$$H_j(B_i) = -\sum_{l=1}^{N_j} \mathsf{P}_j(a_l \in d_{ij}) \cdot \log_2 \mathsf{P}_j(a_l \in d_{ij}) \quad (9)$$

where N_j is the number of different categorical values (a_l) in the j-th attribute, and P_j is defined in (8). It is clear that if we add a new sample to the hyperbox for which most of the sample's categorical values existed in the categorical attributes of the hyperbox, the change in the entropy of that hyperbox

is small. In contrast, if we add a sample into the hyperbox for which most of the sample's categorical values are new symbolic values to the set of existing discrete attributes of the hyperbox, the homogeneity of this hyperbox is significantly changed, and so the entropy will increase. As a result, we can use the change in the entropy of the hypebox as an expansion condition for categorical attributes. This entropy changing value is defined in (10) for each discrete feature j of each hyperbox candidate B_i.

$$Z_j = H_j(B_i \cup \{X\}) - \frac{n_i}{n_i + 1} H_j(B_i) \qquad (10)$$

where $H_j(B_i \cup \{X\})$ is the entropy of the hyperbox B_i on the j-th attribute after covering the input pattern X, computed using (9), n_i is the number of samples contained in the hyperbox B_i (n_i is also equal to the summation of cardinalities of categorical values on the dimension j).

Based on Z_j, we have two ways to construct the expansion condition for categorical attributes:

- The first method is similar to the expansion condition for continuous features. The extended algorithm using this way is denoted by EIOL-GFMM-v1 in this paper. We require the change in the entropy for every categorical attribute smaller than a maximum entropy changing threshold δ for all categorical dimensions:

$$Z_j \le \delta \qquad (11)$$

- The second approach to build the expansion condition for categorical features uses a weaker condition compared to the first way. The proposed online learning algorithm adopting this condition is called EIOL-GFMM-v2 in this paper. This method requires the average change in the entropy of all of the r categorical attributes smaller than a maximum average entropy changing threshold δ:

$$\sum_{j=1}^{r} Z_j \le \delta \qquad (12)$$

If both conditions for categorical and numerical features are met for the hyperbox B_i, it will be temporarily expanded to new coordinates. The expansion of numerical features is performed using (5). Each categorical feature d_{ij} of B_i is expanded as follows:

$$d_{ij}^{new} = \begin{cases} d_{ij}^{old} \cup \{x_j^d : 1\}, & \text{if } \nexists a_j \in d_{ij}^{old} : a_j = x_j^d \\ d_{ij}^{old}.\textbf{update}(a_j), & \text{if } \exists a_j \in d_{ij}^{old} : a_j = x_j^d \end{cases} \qquad (13)$$

where $\textbf{update}(a_j)$ is a function to increase the number of elements of the categorical value a_j by 1. After that, an overlap checking procedure is performed for the newly expanded B_i to examine whether B_i overlaps with any hyperboxes belonging to other classes. In the improved online learning algorithm for numerical features, only four overlap test cases are used as in the original online learning algorithm. However, these four cases are not sufficient to identify all potential overlap cases between two hyperboxes. Therefore, in this extended version, we will deploy a similarity measure between two hyperboxes based on their smallest gap introduced in [6] to check the overlap for numerical features between B_i and other

hyperboxes B_k representing different classes. This similarity measure s_{ik} is defined in (14).

$$s_{ik} = \min_{j=1}^{n}[\min(1 - f(v_{kj} - w_{ij}, 1), 1 - f(v_{ij} - w_{kj}, 1))] \quad (14)$$

where n is the number of continuous features, f is a ramp function given in (3). If B_i and B_k overlap with each other, $s_{ik} = 1$; otherwise, $s_{ik} < 1$. If B_i does not overlap with any B_k representing other classes on the numerical features, we do not need to check the overlap conditions for their discrete features. Otherwise, we have to verify the overlap for the discrete features between B_i and hyperboxes B_k overlapping with B_i on the continuous features. Let Ω_{ij} and Ω_{kj} be the set of categorical values on the j-th discrete attribute of two hyperboxes B_i and B_k, respectively. B_i overlaps with B_k on the j-th categorical feature if and only if:

$$\begin{aligned} &\Omega_j = \Omega_{ij} \cap \Omega_{kj} \ne \varnothing \text{ and} \\ &\exists a_j \in \Omega_j : \mathsf{P}_j(a_j \in d_{ij}) = \mathsf{P}_j(a_j \in d_{kj}) \end{aligned} \qquad (15)$$

where P_j is defined in (8). B_i overlaps with B_k on discrete attributes if the equation (15) is true for all of the r categorical features of these two hyperboxes.

If the hyperbox candidate B_i does not overlap with any hyperboxes B_k representing other classes on either categorical or continuous features, the new coordinates of B_i remain unchanged and the algorithm continues with the next training sample. Otherwise, the coordinates of B_i are reverted to the previous values and the hyperbox candidate with the next highest membership value is selected as an expandable hyperbox candidate, and the above steps are re-iterated.

If none of the hyperbox candidates can be extended to accommodate the input pattern X, a new hyperbox B_i is generated as follows. For each numerical feature j, we set $v_{ij} = x_j^l, w_{ij} = x_j^u, \forall j \in [1, n]$, and for each categorical feature j, we assign $d_{ij} = \{x_j^d : 1\}, \forall j \in [1, r]$.

The classification phase of the EIOL-GFMM algorithm remains unchanged as in the original IOL-GFMM algorithm. We can see that the way of working of EIOL-GFMM algorithm itself can explain the reason leading to the classification results based on the selection of the hyperbox with the maximum membership degree.

B. Properties of the Change in Entropy of Categorical Features when Accommodating New Training Samples

This section presents several interesting properties related to the change of the entropy on each categorical attribute of a hyperbox B_i when accommodating a new training sample X.

Property 1. *When covering an input pattern, the change of the entropy on each discrete attribute j of B_i obtains its maximum value if and only if that attribute j includes a new categorical value which does not exist in the list of its current categorical values. Formally,*

$$d_{ij}^{new} = d_{ij}^{old} \cup \{x_j^d : 1\} \Rightarrow Z_j \mapsto \max \qquad (16)$$

Proof. See Section I in the supplemental material. □

Property 2. *The upper bound of the change in the entropy for every categorical dimension j of B_i depends on the current number of samples included in B_i. That is:*

$$Z_j \leq \log_2(n_i + 1) - \frac{n_i}{n_i + 1} \log_2 n_i \qquad (17)$$

Proof. See Section II in the supplemental material. $\square$

Property 3. *The change of the entropy for each categorical dimension j always falls in the range of $[0, 1]$:*

$$0 \leq Z_j \leq 1; \quad \forall j \in [1, r]$$

Proof. See Section III in the supplemental material. $\square$

Property 3 also confirms that $0 \leq \delta \leq 1$.

Property 4. *When the number of samples contained in B_i approaches infinity, the change of the entropy for every categorical dimension will be limited at 0. Formally,*

$$\lim_{n_i \to +\infty} Z_j = 0, \ \forall j \in [1, r] \qquad (18)$$

Proof. See Section IV in the supplemental material. $\square$

Property 4 indicates that when the number of samples included in each hyperbox B_i increases, the expansion condition for categorical attributes of this hyperbox becomes easier to be satisfied.

IV. EXPERIMENTAL RESULTS

The main purposes of the experiments in this section are to

- Analyze the critical roles of parameters α and δ on classification accuracy for the proposed method
- Compare the performance of the proposed method to relevant approaches of GFMMNN for mixed-attribute data using fixed settings and tuning parameters
- Assess several different methods to estimate the values of α if we have sufficient samples at the training time.

These experiments were conducted on 14 datasets taken from the UCI machine learning repository [1]. These datasets were used in [10] to evaluate different methods to handle mixed-attribute data using the GFMMNN. We used the same datasets to compare our proposed approach to the results presented in that research. The details of these datasets can also be found in subsection V.A in the supplemental document. All of the experimental datasets in this paper are class-imbalanced, so we will use the class balanced accuracy (CBA) metric to assess the performance of classification algorithms. The superior facets of the CBA in comparison to other metrics for the class-imbalanced datasets were shown in [10] and [20].

[1] https://archive.ics.uci.edu/ml/datasets.php

A. Analyzing the Sensitivity of Parameters

There are three important parameters affecting the classification performance of the proposed method, i.e., the maximum hyperbox size for continuous attributes (θ), the trade-off factor (α) regarding the contribution levels of continuous and discrete attributes to the membership function, and the maximum entropy changing threshold (δ) for discrete attributes. The role of θ was analyzed in a recent study [21], in which smaller values of θ usually result in better performance than the use of larger values of θ does. However, the smaller values of θ are, the more complex the final model is (i.e. the larger number of generated hyperboxes). In this section, we only study the influence of two new parameters introduced in our proposed method, i.e., δ and α.

1) Parameter α:

To evaluate the impact of α on the performance of the EIOL-GFMM algorithms, we have changed the values of α from 0 to 1 with step 0.1 and recorded the average CBA scores using 10 times repeated stratified 4-fold cross-validation for 11 mixed-attribute datasets. The impact of α is studied for two cases, i.e., large-sized hyperboxes and small-sized hyperboxes. To obtain the large-sized hyperboxes, we established the parameters $\theta = \delta = 1$ so that the expansion process of hyperboxes is not constrained. To achieve small-sized resulting hyperboxes, we used the small values for both θ and δ, i.e., $\theta = \delta = 0.1$ in this experiment. Fig. 2 shows the change in the CBA for different values of α in the case of large-sized hyperboxes. For $\delta = 1$, the behaviors of the EIOL-GFMM-v1 and EIOL-GFMM-v2 algorithms are identical. Fig. 3 presents the change in the CBA results for different values of α in the case of small-sized hyperboxes for both EIOL-GFMM algorithms. We only present the results for a representative *flag* dataset. The results of the remaining datasets are presented in Figs. S1, S2 and S3 in the supplemental document.

Fig. 2. The change in the class balanced accuracy according to the different values of α for the *flag* dataset ($\theta = 1, \delta = 1$).

In general, the CBA values of both proposed learning algorithms at $\alpha = 0$ (using categorical features only) and $\alpha = 1$ (using numerical features only) are usually smaller than the results of using both types of features. The impact of α on both EIOL-GFMM-v1 and EIOL-GFMM-v2 algorithms are similar for many datasets. It can be observed that the influence of α on the GFMM models with small-sized hyperboxes is significantly higher than that with large-sized hyperboxes. This is demonstrated by the degree of oscillation
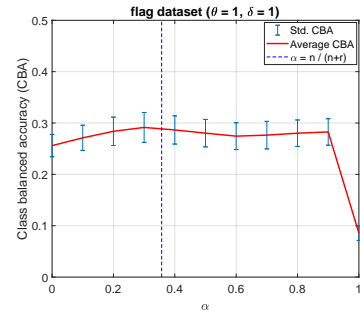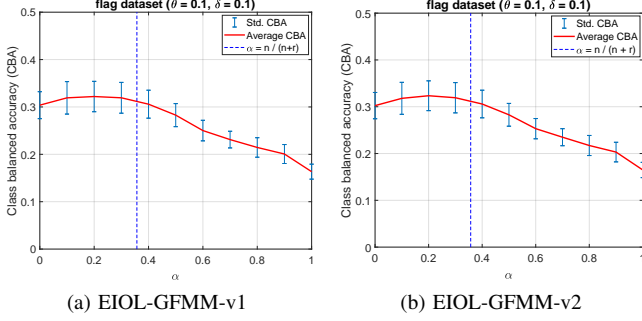
Fig. 3. The change in the class balanced accuracy according to the different values of α for the *flag* dataset ($\theta = 0.1, \delta = 0.1$).

Fig. 4. The change in the class balanced accuracy according to the different values of δ for the *flag* dataset ($\theta = 1, \alpha = n/(n + r)$).

in classification accuracy among different values of α in Fig. 2 and Fig. 3. It is because the value of α affects the results of the membership function, and the membership value in turn impacts the selection of the final hyperbox for each unseen pattern. In the case of small-sized hyperboxes, the number of hyperboxes is high and the small change in the membership value can lead to a significant change in the selected hyperbox.

As can be seen, the selection of α can result in the change in the classification performance, thus this parameter needs to be tuned in the learning process. However, for a large number of training samples, performing a hyper-parameter tuning step for α is time-consuming. For the online learning process, we also face another scenario, where we do not have sufficient samples at the training time. In this case, we cannot conduct the tuning process using a cross-validation technique to select an appropriate α for the learning algorithms. Therefore, we usually use a fixed setting for α. From the empirical results in Figs. 2, 3 and Figs. S1, S2, and S3 in the supplemental document, it is interesting to observe that the highest CBA results are usually obtained for the value of α near the threshold $n/(n + r)$. Therefore, in the case of using a fixed setting for α, we will set $\alpha = n/(n+r)$. With this setting, each feature is treated as equally important in decision making.

2) Parameter δ:

In this subsection, we will assess the impact of the maximum entropy changing threshold (δ) for discrete attributes on the classification performance. To rule out the influence of θ, we set $\theta = 1$ so that numerical features can be expanded without any limitation. From the above experimental results, we used $\alpha = n/(n + r)$. Therefore, the performance of the learning algorithms depends on the selection of δ. We changed δ from 0.05 to 0.1 and kept the change step of 0.1 up to 1. The impact of δ on the IOL-GFMM-v1 and IOL-GFMM-v2 algorithms is illustrated in Fig. 4 for the *flag* dataset. The results for the remaining datasets can be found in Figs. S4 and S5 in the supplemental document.

From these results, it can be observed that the change in the CBA results for the EIOL-GFMM-v1 using $\delta < 0.7$ is very small. For $\alpha \geq 0.7$, its impact on the classification performance of the EIOL-GFMM-v1 on a number of datasets such as *autralian*, *heart*, and *post operative* is significant, especially in the case of $\delta = 1$. In general, the classification error for $\delta \geq 0.7$ in the EIOL-GFMM-v1 is relatively high.
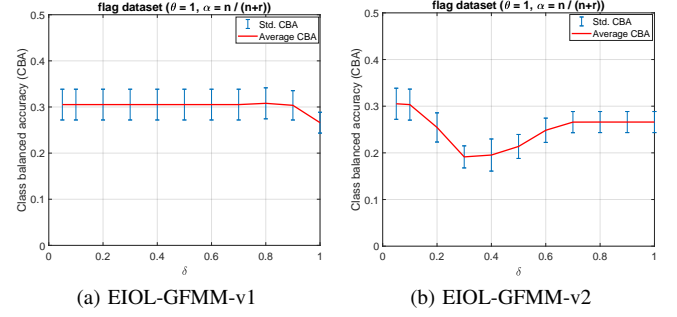
For the EIOL-GFMM-v2, however, the change in CBA values is small for $\delta < 0.2$ and $\delta \geq 0.7$. In contrast, the classification performance has significantly changed for the values of δ from 0.2 to 0.7. It can be seen that the impact of δ on the performance of the EIOL-GFMM-v2 algorithm is higher than that on the EIOL-GFMM-v1 algorithm. It is because the hyperbox expansion procedure for discrete features in the EIOL-GFMM-v2 algorithm can be performed much easier than that in the EIOL-GFMM-v1 algorithm. As a result, the number of generated hyperboxes in the EIOL-GFMM-v1 algorithm is higher than that of hyperboxes in the EIOL-GFMM-v2 algorithm, and so it can capture better the underlying data distribution. For the EIOL-GFMM-v1 algorithm, each discrete feature can only accommodate a new categorical value if the number of samples for the current categorical values is sufficiently large according to Properties 1 and 2. As a result the homogeneity for each categorical feature of hyperboxes in the EIOL-GFMM-v1 is higher compared to that in the EIOL-GFMM-v2. Therefore, the change in the classification performance among the different values of δ in the first version of the proposed method is smaller in comparison to the second version.

B. Comparing the Performance of the EIOL-GFMM Algorithms with Other Methods using the Fixed-Parameter Settings

As we assume that there will not be sufficient number of training samples up front in the considered online learning scenarios, as discussed earlier, we will use a fixed setting for certain hyper-parameters which cannot be reliably tuned/optimised using available data. This section is to assess the proposed method in comparison to other solutions to deal with mixed-attribute data for the GFMMNN shown in [10] using previously evaluated fixed values of hyper-parameters. In particular, the proposed method will be compared with two learning algorithms with the mixed-attribute handling ability for the GFMM model including the Onln-GFMM-M1 [15] and the Onln-GFMM-M2 [16]. We will also compare the proposed method to the use of the original IOL-GFMM algorithm along with various encoding methods for categorical features.

1) Algorithms with Mixed-Attribute learning ability:

In [10], the different learning methods were compared to each other using three different settings for θ, i.e., a small size $\theta = 0.1$, a large size $\theta = 0.7$, and an extreme case $\theta = 1$.

To compare the proposed method to the previous solutions, we also used the same settings for the θ parameter. For the γ parameter, we set $\gamma = 1$ as recommended in [22]. In addition to θ and γ, the existing learning algorithms for the GFMMNN with mixed-feature handling ability have their own hyper-parameters. The Onln-GFMM-M1 algorithm depends on the η parameter, which represents the maximum hyperbox size for discrete features. We used $\eta \in \{0.1, 0.7, 1\}$ as shown in [10]. The Onln-GFMM-M2 algorithm has the β parameter to control the minimum number of categorical features matched between the selected hyperbox and the input pattern so that hyperbox can be expanded to cover the input pattern. Similarly to [10], we used $\beta \in \{25\%, 50\%, 75\%\}$ of the total number of features for each dataset. To be fair in the comparison, we used the parameter $\delta \in \{0.1, 0.7, 1\}$ and $\alpha = n/(n+r)$ for the proposed learning algorithms.

The average CBA values over 10 times repeated stratified 4-fold cross-validation with different parameter settings are shown in Table S.II in the supplemental document. It can be easily observed that in extreme cases (the largest values of parameters), the classification performance of our proposed method significantly outperforms the Onln-GFMM-M1 and Onln-GFMM-M2 algorithms. To facilitate the comparison of results, for each value of θ, we will use the best results of the remaining parameter to rank four algorithms over 14 datasets. For example, for $\theta = 0.7$, the Onln-GFMM-M1 usually obtains the best performance using $\eta = 0.1$, the Onln-GFMM-M2 achieve its best results with $\beta = 0.75r$, and two proposed methods attain their best results using $\delta = 0.1$. The average ranking of algorithms using their best settings is shown in Table I.

TABLE I
THE AVERAGE RANK FOR THE ALGORITHMS USING THEIR BEST SETTINGS

θ	Method	Other parameters	Average rank
0.1	Onln-GFMM-M1	$\eta = 0.1$	3
	Onln-GFMM-M2	$\beta = 0.75r$	3.214
	EIOL-GFMM-v1	$\delta = 0.1$	1.893
	EIOL-GFMM-v2	$\delta = 0.1$	1.893
0.7	Onln-GFMM-M1	$\eta = 0.1$	2.429
	Onln-GFMM-M2	$\beta = 0.75r$	3.857
	EIOL-GFMM-v1	$\delta = 0.1$	1.607
	EIOL-GFMM-v2	$\delta = 0.1$	2.107
1	Onln-GFMM-M1	$\eta = 0.1$	2.429
	Onln-GFMM-M2	$\beta = 0.75r$	3.857
	EIOL-GFMM-v1	$\delta = 0.1$	1.679
	EIOL-GFMM-v2	$\delta = 0.1$	2.036

It can be seen that the classification performance of our proposed methods is better than that of two existing algorithms with the mixed-attribute learning ability for three different thresholds of θ. To conclude if there are statistically significant differences among algorithms, we will carry out a non-parametric test procedure as recommended in [23] employing the Friedman rank-sum test with a confidence level of 95% (a significance level $\epsilon = 0.05$). The null hypothesis is "there are no statistical differences between learning algorithms", and if this hypothesis is rejected, then we perform the Nemenyi post-hoc test to determine the particular differences. For Z datasets and M algorithms, the Friedman statistic distribution is computed using average ranks $R_j (j \in [1, M])$ of each algorithm j as follows:

$$\chi_F^2 = \frac{12Z}{M \cdot (M+1)} \left[\sum_{j=1}^{M} R_j^2 - \frac{M \cdot (M+1)^2}{4} \right] \quad (19)$$

From χ_F^2, a F-distribution with $M-1$ and $(M-1) \cdot (N-1)$ degrees of freedom can be calculated using (20).

$$F_F = \frac{(Z-1) \cdot \chi_F^2}{Z \cdot (M-1) - \chi_F^2} \quad (20)$$

The rejection of the null hypothesis occurs with the significance level ϵ if F_F is smaller than a critical value of $F(M-1, (M-1) \cdot (N-1), \epsilon)$. In this experiment, we used 14 datasets and four learning algorithms, so F_F is distributed according to the F-distribution with $4 - 1 = 3$ and $(4-1) \cdot (14-1) = 39$ degrees of freedom. The critical value of $F(3, 39)$ for $\epsilon = 0.05$ is 2.845.
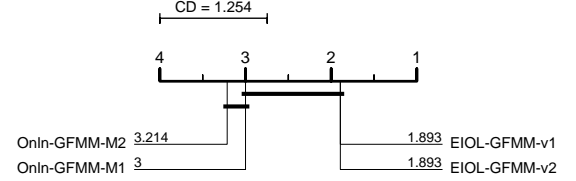
Fig. 5. Critical difference diagram for four learning algorithms ($\theta = 0.1$).

For $\theta = 0.1$, we obtain $F_F = 5.5539 > 2.845$, and so the null hypothesis is rejected. This means that there are significant differences between the results of learning algorithms. Using the Nemenyi post-hoc test, we obtain a critical difference (CD) diagram in Fig. 5. The groups of algorithms that are not significantly different from each other are connected by a solid line. We can see that our proposed methods are statistically better compared to the Onln-GFMM-M2 algorithm with the selected settings. However, there is no statistically significant difference in the classification performance between the proposed methods and the Onln-GFMM-M1 algorithm.
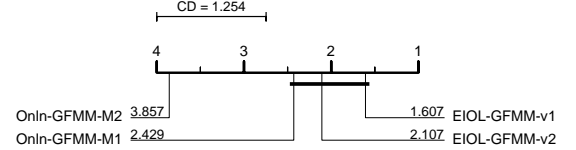
Fig. 6. Critical difference diagram for four learning algorithms ($\theta = 0.7$).

For $\theta = 0.7$, we have $F_F = 16.5238 > 2.845$, and so the null hypothesis is also rejected. Applying the Nemenyi post-hoc test, we have a CD diagram in Fig. 6. We can observe that there are no statistically significant differences among the obtained empirical results in the groups of two proposed learning algorithms and the Onln-GFMM-M1 algorithm. However, the algorithms in this group significantly outperform the Onln-GFMM-M2 algorithm.

Similarly, for $\theta = 1$, we obtain $F_F = 15.7716 > 2.845$, and so the null hypothesis is rejected as well. Fig. 7 shows the CD diagram using the Nemenyi post-hoc test. In this case, the
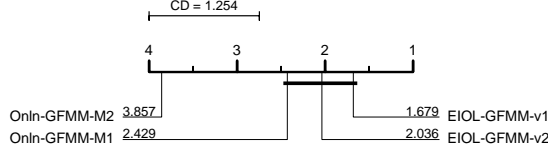
Fig. 7. Critical difference diagram for four learning algorithms ($\theta = 1$).

statistical difference in the classification performance among the four methods is the same as in the case of $\theta = 0.7$.

For the complexity of the resulting GFMM models using these learning algorithms, the average number of generated hyperboxes for each method is shown in Table S.III in the supplemental document. We can see that in most of the cases, the complexity of the model using the Onln-GFMM-M2 algorithm is lowest, while the complexity of the GFMMNN using the EIOL-GFMM-v1 is highest. The number of generated hyperboxes using the EIOL-GFMM-v2 algorithm is usually smaller than that using the Onln-GFMM-M1 algorithm.

2) Comparing the Proposed Method to the Original Learning Algorithm using Encoding Methods:

In this subsection, we will compare the EIOL-GFMM algorithms to the original IOL-GFMM algorithm using different encoding techniques. In [10], there are eight encoding methods used to transform the categorical features into numerical features, i.e., Leave-One-Out (LOO), CatBoost, label, one-hot, target, James-Stein, Helmert, and Sum encoding techniques. Similarly to the above experiment, we will consider three different thresholds for θ including 0.1, 0.7, and 1. To be fair in the comparison, we established the value of δ equal to θ.

The average CBA values over 10 times repeated stratified 4-fold cross-validation of these approaches are shown in Table S.IV, and their ranks are presented in Table S.V in the supplemental document. The average rank for these methods for different thresholds of θ is shown in Table II, in which the best results are highlighted in bold. In general, we can see that the average performance of our proposed method is better than that using the original IOL-GFMM algorithm along with encoding techniques. One of the strong points of our proposed method is that it does not use any encoding method for discrete attributes.

TABLE II
THE AVERAGE RANKS FOR THE PROPOSED METHOD AND THE ORIGINAL IOL-GFMM USING DIFFERENT ENCODING TECHNIQUES

Method	$\theta(= \delta)$		
	0.1	0.7	1
IOL-GFMM + CatBoost	5.357	5.071	5
IOL-GFMM + One-hot	8.179	8.036	7.536
IOL-GFMM + LOO	4.857	4.786	5.643
IOL-GFMM + Label	5.107	5.607	5.464
IOL-GFMM + Target	5.464	4.464	5.429
IOL-GFMM + James-Stein	5.250	4.536	5.214
IOL-GFMM + Helmert	7.893	7.750	7.179
IOL-GFMM + Sum	5.607	5.750	7.821
EIOL-GFMM-v1	3.679	**3.536**	**2.857**
EIOL-GFMM-v2	**3.607**	5.464	**2.857**

Similarly to the above experiments, we will use a statistical test procedure to analyze the statistical difference among the

methods. The critical value of $F(9, 117)$ for 10 methods and 14 datasets at $\epsilon = 0.05$ is 1.9608.

For $\theta(= \delta) = 0.1$, we obtain $F_F = 4.289 > 1.9608$, and so there are statistically significant differences among methods. Using the Nemenyi post-hoc test, we obtain a CD diagram in Fig. 8. We can see that the proposed method is significantly better than the original IOL-GFMM algorithm using the one-hot or Helmert encoding method. However, there are no statistical differences between the proposed methods and the IOL-GFMM algorithm using the remaining encoding techniques.

Fig. 8. Critical difference diagram for the proposed method and the original algorithm using encoding methods ($\theta = 0.1$).

For $\theta(= \delta) = 0.7$, we have $F_F = 3.6596 > 1.9608$, and so there are also statistically significant differences among methods in this case. Employing the Nemenyi post-hoc test, we obtain a CD diagram in Fig. 9. In this case, the EIOL-GFMM-v1 is significantly better than the original IOL-GFMM algorithm using the one-hot or Helmert encoding method as well, but it does not statistically outperform the original algorithm using the remaining encoding approaches. Moreover, in this case, there is not sufficient evidence to conclude that the EIOL-GFMM-v2 is statistically better than the original algorithm employing encoding methods.

Fig. 9. Critical difference diagram for the proposed method and the original algorithm using encoding methods ($\theta = 0.7$).

Fig. 10. Critical difference diagram for the proposed method and the original algorithm using encoding methods ($\theta = 1$).

For the extreme case $\theta(=\delta)=1$, we obtain $F_F = 6.2138 > 1.9608$, and so the null hypothesis is also rejected. Fig. 10 shows the CD diagram, in this case, usin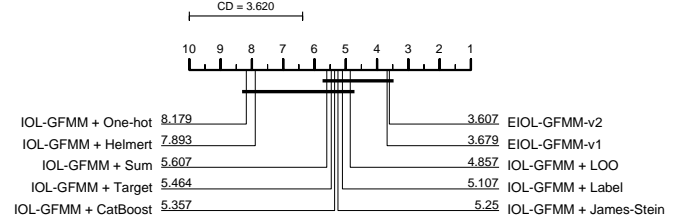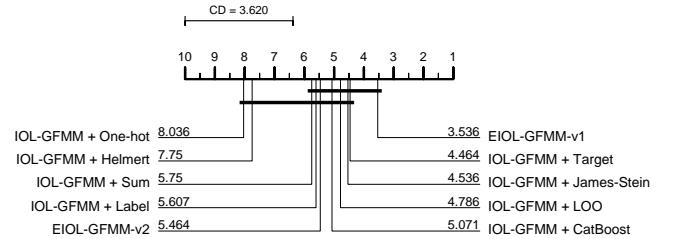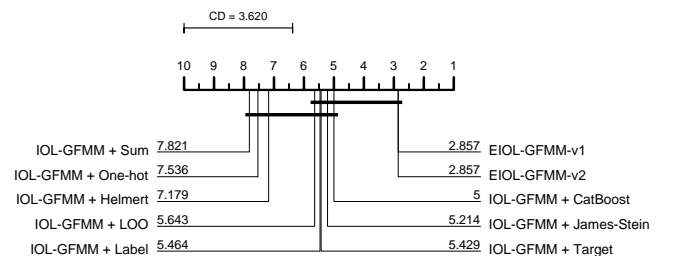g the Nemenyi post-hoc test. It can be observed that there are no statistical differences among the original algorithms using encoding methods. Our proposed methods significantly outperform the original learning algorithm using the one-hot, sum, or Helmert encoding method. However, there are no statistical differences between our proposed methods and the original IOL-GFMM algorithm using the remaining encoding approaches.

C. Evaluating the Role of the Hyper-Parameter Tuning on the Performance of the EIOL-GFMM Algorithms

1) Hyper-Parameter Tuning and the Estimation of α:

In the case that we are given a large number of samples at the training time to build an initial model, we can perform a hyperparameter tuning step for α but this process is time-consuming. Meanwhile, the empirical results in subsection IV-A1 indicate the relation between suitable values of α and the ratio of the number of continuous features over the total number of features. In this subsection, we propose a simple way to estimate the appropriate value of α in a data-driven manner. The estimation method does not loop through predefined values of α as in the tuning process, and so they will run faster than the hyper-parameter tuning step for α.

For each training fold T_i, we split it into three inner folds. To estimate the value of α for each training fold T_i, we will repeat the learning process three times. Each time two inner folds are used to build the GFMM model and the remaining inner fold is used as a validation set. For each inner training fold $T_{ij}(j \in [1,3])$, we split it into two separate parts, in which each part contains either continuous attributes or discrete attributes. Then, we will construct two separate GFMM models using the EIOL-GFMM algorithm from these two training parts. After that, we compute the CBA value for each trained model using the inner validation fold V_{ij}. Let CBA_{ij1} and CBA_{ij2} be the CBA scores for the GFMM models trained on continuous features only and discrete features only, respectively. We have two ways to estimate the value of α for each training fold T_i. The first way uses both the CBA values and the number of features, denoted Est-α-v1 in this paper, as follows:

$$\hat{\alpha} = \frac{\sum\limits_{j=1}^{3} CBA_{ij1} \cdot n}{\sum\limits_{j=1}^{3} CBA_{ij1} \cdot n + \sum\limits_{j=1}^{3} CBA_{ij2} \cdot r} \quad (21)$$

The second estimation way of α uses only the obtained CBA values, called Est-α-v2, as follows:

$$\hat{\alpha} = \frac{\sum\limits_{j=1}^{3} CBA_{ij1}}{\sum\limits_{j=1}^{3} CBA_{ij1} + \sum\limits_{j=1}^{3} CBA_{ij2}} \quad (22)$$

Two ways of estimating α are summarized in Fig. S6 in the supplemental material. After obtaining the value of α, we use it to train a final model using the whole mixed-attribute training fold T_i and evaluate its performance using the i-th testing fold. The above process is repeated 40 times (10 times repeated stratified 4-fold cross-validation) to compare the average CBA values among different methods.

This section will compare the effectiveness of two above estimation methods to the fixed setting of $\alpha = n/(n+r)$ and the parameter tuning method for α. In the parameter tuning method, for each training fold T_i, we split T_i into three inner training folds. Two inner training folds are used to build the GFMM model using the proposed EIOL-GFMM algorithm and the remaining fold is used as a validation fold. We will iterate this process three times to obtain three CBA values from three validation folds for each value $\alpha \in \{0, 0.1, \ldots, 0.9, 1\}$. The value of α resulting in the highest average CBA value over three inner validation folds is used to build the final GFMM model on the whole training fold T_i, and the trained model is assessed by the CBA value on the i-th testing fold. The whole process is repeated 40 times for different training folds T_i.

The average CBA results of 40 GFMM models trained using the proposed algorithms with two estimation methods of α, the parameter tuning method and the fixed setting of α for 11 datasets are shown in Table S.VI in the supplemental material. The rank for these methods is presented in Table S.VII in the supplemental document. It is noted that the results are reported over 11 out of 14 experimental datasets because these datasets contain both continuous and discrete features while three remaining datasets consist of only discrete features. The average rank over 11 datasets for different methods of finding the value of α for the GFMM model trained using the proposed algorithm is shown in Table III. Similarly to subsection IV-A1, we compare the methods of finding α in two cases, i.e., small-sized hyperboxes ($\theta = \delta = 0.1$) and large-sized hyperboxes ($\theta = \delta = 1$). The best rank in each row is highlighted in bold. In the case of $\theta = \delta = 1$, the behavior of both EIOL-GFMM-v1 and EIOL-GFMM-v2 is the same, and so they lead to the same results.

TABLE III
AVERAGE RANK FOR DIFFERENT METHODS USED TO FIND THE VALUES FOR PARAMETER α

Algorithm	$\theta = \delta$	Tuning α	Est-α-v1	Est-α-v2	$\alpha = n/(n+r)$
EIOL-GFMM-v1	0.1	2.727	**2**	3	2.273
EIOL-GFMM-v2	0.1	2.545	**2.318**	2.727	2.409
Both	1	**2.273**	2.773	**2.273**	2.682

We can observe that for small values of θ and δ, the estimation method using the CBA values from two separate models along with the number of features usually results in the best average CBA values in comparison to the second estimation method, the parameter tuning approach, and the fixed setting of α for both learning algorithms. However, the second estimation method without using the number of features often leads to the worst results. Interestingly, in this case, the fixed value of $\alpha = n/(n+r)$ shows slightly better results than the hyper-parameter tuning method. In the case of generating the largest hyperbox sizes, the best predictive results belong to the models using the hyper-parameter tuning method and the Est-α-v2 method. Meanwhile, the first estimation method usually

leads to the worst classification performance.

To explain these facts, we will examine the distribution of the obtained values of α through 40 iterations and the change in the corresponding CBA values. Fig. 11 shows the distribution of the obtained α values for different methods in the case of largest-sized hyperboxes for the *flag* dataset. The results in the case of $\theta = \delta = 0.1$ are presented in Fig. 12. Similar results for all of the remaining datasets can be found in Figs. S7, S8, and S9 in the supplemental material.
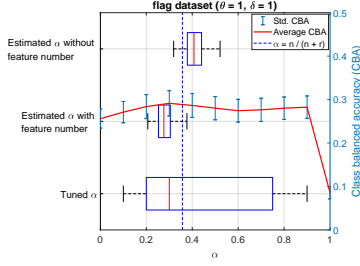
Fig. 11. The distribution of the obtained α values for different methods used to find α and the CBA values for the *flag* dataset ($\theta = 1, \delta = 1$).

(a) EIOL-GFMM-v1   (b) EIOL-GFMM-v2
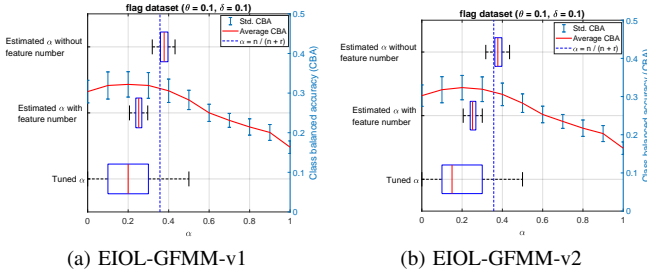
Fig. 12. The distribution of the obtained α values for different methods used to find α and the CBA values for the *flag* dataset ($\theta = 0.1, \delta = 0.1$).

We can see that, in both cases, the use of the hyper-parameter tuning method returns a wide range of values for α, in which the obtained median value of α locates near the α value resulting in the best classification result. In the case of small-sized hyperboxes, it can be seen that the deviation in the classification results among adjacent values of α is high. Therefore, a wide range of α values usually leads to a low average classification result compared to the use of a narrow range of α values near the best results. We can see from Fig. 12, the obtained α values employing two estimation methods are distributed in a narrower area than that using the hyper-parameter tuning approach. Also, the range of the obtained α values of the Est-α-v2 is wider than that using the Est-α-v1 method. However, the range of the obtained α values using the Est-α-v1 is nearer the α value leading to the best classification performance than one using the Est-α-v2. Hence, in this case, the Est-α-v1 method usually gives the best classification results among the four methods.

In the case of largest-sized hyperboxes, the difference in the performance among different values of α is small. Therefore, the wide range of the obtained α values using the hyper-parameter tuning method regularly leads to a better average classification result compared to the outcomes employing other

methods. As can be seen from Fig. S7 in the supplemental document that two estimation methods return a narrower range of the obtained α values in comparison to the use of the hyper-parameter tuning approach. However, in this case, the obtained α values using the Est-α-v2 usually locates nearer the α values leading to much better performance than those in the case of using the Est-α-v1 method. Therefore, the performance of the GFMM model using the Est-α-v2 outperforms that adopting the Est-α-v1 method.

In short, the second estimation method is appropriate for the model having a small number of hyperboxes, while the first estimation method should be used in the case when the resulting model has a large number of hyperboxes.

2) Comparing the EIOL-GFMM Algorithms to Other Algorithms with the Mixed-Attribute Learning Ability:

In this experiment, we will compare the classification performance of our proposed method and two existing algorithms with the mixed-attribute learning ability using the hyper-parameter tuning procedure for important parameters in each learning algorithms. For the θ value in all learning algorithms, we will find its best parameter value in the range of $\{0.1, 0.2, \ldots, 0.9, 1\}$ for each training fold. The η parameter for the Onln-GFMM-M1 algorithm is searched in the range of $\{0.1, 0.3, 0.5, 0.9, 1\}$. The searching range of the β parameter for the Onln-GFMM-M2 is $\{10\%, 30\%, 50\%, 70\%, 90\%, 100\%\}$ of the total number of categorical features. For the two proposed algorithms in this paper, the δ parameter is searched in the range of $\{0.1, 0.3, 0.5, 0.9, 1\}$, while the α value is sought in the range of $\{0, 0.1, \ldots, 0.9, 1\}$.

Each training fold T_i is split into three inner folds, in which two inner folds are used for training a GFMM model using learning algorithms. Then, we use the remaining fold to obtain the CBA value. This process is repeated three times for every inner validation fold. The combination of parameters resulting in the best average CBA values through three validation folds is used to train the final GFMM model using the whole training fold T_i. After that, this model is evaluated using the corresponding testing fold. This process is iterated 40 times (10 times repeated stratified 4-fold cross-validation) for each dataset. The average CBA results for four learning methods using the above hyper-parameter tuning approach are shown in Table S.VIII and their ranks are presented in Table S.IX in the supplemental document. The average rank for each method over 11 mixed-attribute datasets is shown in Table IV.

TABLE IV
AVERAGE RANKS FOR THE LEARNING ALGORITHMS USING THE
HYPER-PARAMETER TUNING APPROACH

Algorithm	Average rank
Onln-GFMM-M1	3.182
Onln-GFMM-M2	2.909
EIOL-GFMM-v1	**1.5**
EIOL-GFMM-v2	2.409

We can observe that the two proposed learning algorithms outperform two existing learning algorithms with the mixed-attribute handling ability, in which the best performance belongs to the EIOL-GFMM-v1 algorithm. For the experimental

results in subsection IV-B1, we can see that the Onln-GFMM-M1 algorithm is better than the Onln-GFMM-M2 algorithm using the fixed-parameter settings. However, by using the hyper-parameter tuning method, the Onln-GFMM-M2 algorithm overcomes the Onln-GFMM-M1. This is because of the difference in the distribution between the inner training set used to find the best combination of parameters and the training fold used to build the final model. The Onln-GFMM-M1 needs to use the entire training data to find the distance between categorical values based on the relationship between the occurrence frequency of discrete values and classes. These distance values are deployed to build membership functions. Therefore, when the training data change, the best combination of parameters on the inner training folds no longer maintains the superior classification performance when used on the training fold T_i. Our proposed methods do not use the training samples to build the similarity measure among categorical features, and so they still achieve the best performance as in the case of using the fixed parameter settings.

Interestingly, the classification performance of learning algorithms using the hyper-parameter tuning method in several datasets such as *cmc*, *cmc*, *zoo*, *australian*, and *japanese credit* is worse than those using fixed parameter settings presented in subsection IV-B1. This is because the representativeness and distribution of the inner validation sets used to find the best combination of parameters are different from the training and testing folds. Therefore, the best parameters obtained from the inner validation folds may not lead to the best classification accuracy on the testing set. As a result, the hyper-parameter tuning method does not always result in better performance than the use of fixed parameters.

To verify the statistical difference in the performance among the learning algorithms, we will use the above Friedman rank-sum test. For 11 datasets and 4 learning algorithms, F_F is distributed according to the F-distribution with $4 - 1 = 3$ and $(4 - 1) \cdot (11 - 1) = 30$ degrees of freedom. The critical value of $F(3, 30)$ at a significant level $\epsilon = 0.05$ is 2.9223. In this case, we obtain $F_F = 4.884 > 2.9223$. Therefore, there are statistically significant differences among the four considering algorithms. Using the Nemenyi post-hoc test, we achieve a CD diagram in Fig. 13.
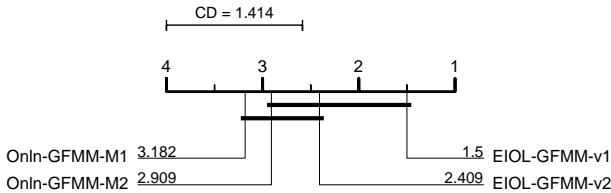
Fig. 13. A CD diagram of four learning algorithms using the hyper-parameter tuning method.

We can see that there is a statistically significant difference in the classification performance between the EIOL-GFMM-v1 and the Onln-GFMM-M1 algorithms in this case. For $CD = 1.414$, we can also conclude that the EIOL-GFMM-v1 algorithm significantly better than the Onln-GFMM-M2 algorithm. However, the EIOL-GFMM-v2 does not statistically

outperform both existing learning algorithms with the mixed-attribute learning ability.

V. Conclusion and Future Work

This paper presented a new online learning algorithm for the GFMMNN with mixed-attribute data. The proposed method expands the current membership function for both continuous and discrete features. We also extend the current architecture of the GFMM model for mixed-attribute data and introduce a new way of learning for categorical dimensions based on the change in the entropy when accommodating new discrete values without using any encoding methods. The experimental results confirmed the superior classification performance of our proposed method in comparison to the current solutions to handle the mixed-type datasets for the GFMMNN.

Although the GFMMNN for mixed-attribute data itself can explain the predicted results using the membership function to select the appropriate hyperbox, to make it friendly and easy-to-read for users, it is necessary to extract and optimize *if-then* rule sets from the resulting hyperboxes for both continuous and discrete features in the future studies. The interpretability of predictive models is a critical factor when applying the machine learning algorithms for high-stakes applications such as medicine, finance, or criminal justice [24]. Furthermore, the classification accuracy depends on the selection of parameters, thus the next research should assess the use of optimization algorithms such as genetic algorithms [25] to evolve the hyperboxes and optimize their hyperparameters simultaneously. When applying the online learning algorithms for applications in dynamic changing environments, these learning algorithms need to detect and adapt to the change of the underlying data distribution [4], [7], [26], [27], [28]. Therefore, one of the potential research directions is to integrate the adaptation ability into the proposed algorithm.

References

[1] B. Lakshminarayanan, D. M. Roy, and Y. W. Teh, "Mondrian forests: Efficient online random forests," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, 2014, p. 31403148.

[2] E. Lughofer and M. Pratama, "Online active learning in data stream regression using uncertainty sampling based on evolving generalized fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 1, pp. 292–309, 2018.

[3] B. Gabrys and A. Bargiela, "General fuzzy min-max neural network for clustering and classification," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 769–783, 2000.

[4] B. Gabrys and A. Bargiela, "Neural networks based decision support in presence of uncertainties," *Journal of Water Resources Planning and Management*, vol. 125, pp. 272–280, 1999.

[5] P. K. Simpson, "Fuzzy min-max neural networks. i. classification," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 776–786, 1992.

[6] B. Gabrys, "Agglomerative learning algorithms for general fuzzy min-max neural network," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 32, no. 1, pp. 67–82, 2002.

[7] ——, "Learning hybrid neuro-fuzzy classifier models from data: to combine or not to combine?" *Fuzzy Sets and Systems*, vol. 147, no. 1, pp. 39–56, 2004.

[8] A. Bargiela, W. Pedrycz, and M. Tanaka, "An inclusion/exclusion fuzzy hyperbox classifier," *International Journal of Knowledge-based and Intelligent Engineering Systems*, vol. 8, no. 2, pp. 91–98, 2004.

[9] T. T. Khuat, F. Chen, and B. Gabrys, "An improved online learning algorithm for general fuzzy min-max neural network," in *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–9.

[10] T. T. Khuat and B. Gabrys, "An in-depth comparison of methods handling mixed-attribute data for general fuzzy min-max neural network," *arXiv e-prints*, p. arXiv:2009.00237, 2020.

[11] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS18, 2018, p. 66396649.

[12] R. K. Brouwer, "A feed-forward network for input that is both categorical and quantitative," *Neural Networks*, vol. 15, no. 7, pp. 881 – 890, 2002.

[13] T. Huang, Y. He, D. Dai, W. Wang, and J. Z. Huang, "Neural network-based deep encoding for mixed-attribute data classification," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2019, pp. 153–163.

[14] T. T. Khuat, D. Ruta, and B. Gabrys, "Hyperbox based machine learning algorithms: A comprehensive survey," *Soft Computing*, pp. 1–39, 2020.

[15] P. R. D. Castillo and J. Cardenosa, "Fuzzy min-max neural networks for categorical data: application to missing data imputation," *Neural Computing and Applications*, vol. 21, no. 6, pp. 1349–1362, 2012.

[16] S. Shinde and U. Kulkarni, "Extracting classification rules from modified fuzzy minmax neural network for data with mixed attributes," *Applied Soft Computing*, vol. 40, pp. 364 – 378, 2016.

[17] B. Gabrys, "Combining neuro-fuzzy classifiers for improved generalisation and reliability," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 3, 2002, Conference Proceedings, pp. 2410–2415.

[18] T. T. Khuat and B. Gabrys, "Accelerated learning algorithms of general fuzzy min-max neural network using a novel hyperbox selection rule," *Information Sciences*, vol. 547, pp. 887–909, 2021.

[19] T. Li, S. Ma, and M. Ogihara, "Entropy-based criterion in categorical clustering," in *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, 2004, pp. 68–75.

[20] L. Mosley, "A balanced approach to the multi-class imbalance problem," Ph.D. dissertation, Iowa State University, 2013.

[21] T. T. Khuat and B. Gabrys, "A comparative study of general fuzzy min-max neural networks for pattern classification problems," *Neurocomputing*, vol. 386, pp. 110 – 125, 2020.

[22] S. Abe, "Dynamic fuzzy rule generation," in *Pattern Classification: Neuro-fuzzy Methods and Their Comparison*. Springer London, 2001, pp. 177–196.

[23] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[24] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.

[25] T. T. Khuat and M. H. Le, "A genetic algorithm with multi-parent crossover using quaternion representation for numerical function optimization," *Applied Intelligence*, vol. 46, no. 4, pp. 810–826, 2017.

[26] Z. Sahel, A. Bouchachia, B. Gabrys, and P. Rogers, "Adaptive mechanisms for classification problems with drifting data," in *Proc. of the 11th International Conference on Knowledge-based Intelligent Engineering Systems (KES'2007)*. Springer, 2007, pp. 419–426.

[27] P. Kadlec and B. Gabrys, "Architecture for development of adaptive online prediction models," *Memetic Computing*, vol. 1, no. 4, pp. 241–269, 2009.

[28] M. Salvador, M. Budka, and B. Gabrys, "Effects of change propagation resulting from adaptive preprocessing in multicomponent predictive systems," *Procedia Computer Science*, vol. 96, pp. 713–722, 2016.

Supplemental Material for the Paper:
An Online Learning Algorithm for a Neuro-Fuzzy
Classifier with Mixed-Attribute Data

Thanh Tung Khuat [ID], *Student Member, IEEE*, and Bogdan Gabrys [ID], *Senior Member, IEEE*

I. PROOF OF PROPERTY 1 FROM THE MAIN PAPER

Property 1. *When covering an input pattern, the change of the entropy on each discrete attribute j of B_i obtains its maximum value if and only if that attribute j includes a new categorical value which does not exist in the list of its current categorical values. Formally,*

$$d_{ij}^{new} = d_{ij}^{old} \cup \{x_j^d : 1\} \Rightarrow Z_j \mapsto \max$$

Proof. From the main paper, the current entropy of categorical values for the discrete feature j is computed as follows:

$$H_j(B_i) = -\sum_{l=1}^{N_j} \mathsf{P}_j(a_l \in d_{ij}) \cdot \log_2 \mathsf{P}_j(a_l \in d_{ij})$$

We have:

$$\mathsf{P}_j(a_l \in d_{ij}) = \frac{|\{a \in d_{ij} | a = a_l\}|}{|d_{ij}|} = \frac{\varphi(a_l)}{n_i}$$

where $\varphi(a_l)$ is a function returning the number of elements of a_l on the discrete dimension d_{ij} of B_i and n_i is the number of samples included in the hyperbox B_i. Therefore,

$$H_j(B_i) = -\sum_{l=1}^{N_j} \frac{\varphi(a_l)}{n_i} \cdot \log_2 \frac{\varphi(a_l)}{n_i}$$

Case 1: The j-*th* discrete feature includes a new categorical value x_j^d which does not exist in the current list of categorical values of d_{ij}.

In this case, the number of categorical values on the discrete attribute j after including x_j^d is $N_j + 1$ and $\varphi(a_l)(l \in [1, N_j]$ is unchanged. The number of samples contained in B_i is $n_i + 1$. As a result, we obtain: $\mathsf{P}_j(x_j^d \in d_{ij}) = \dfrac{1}{n_i + 1}$

Now, the new entropy on the discrete feature j when including x_j^d is:

$$H_j^{(1)}(B_i \cup \{X\}) = -\sum_{l=1}^{N_j} \mathsf{P}_j(a_l \in d_{ij}) \cdot \log_2 \mathsf{P}_j(a_l \in d_{ij}) - \mathsf{P}_j(x_j^d \in d_{ij}) \cdot \log_2 \mathsf{P}_j(x_j^d \in d_{ij})$$

$$= -\sum_{l=1}^{N_j} \frac{\varphi(a_l)}{n_i + 1} \cdot \log_2 \frac{\varphi(a_l)}{n_i + 1} - \frac{1}{n_i + 1} \cdot \log_2 \frac{1}{n_i + 1}$$

Case 2: The j-*th* discrete feature includes a categorical value x_j^d existed in the current list of categorical values of d_{ij}.

In this case, the number of distinct categorical values, N_j, storied on each of the j-*th* discrete dimension is unchanged, while the number of samples included in B_i increases by 1. Without loss of generality, we assume that $x_j^d = a_{N_j}$, then $\varphi(a_l)$ $(l \in [1, N_j - 1]$ is unchanged, while $\varphi(x_j^d) = \varphi^{new}(a_{N_j}) = \varphi^{old}(a_{N_j}) + 1$. Therefore, $\mathsf{P}_j(x_j^d \in d_{ij}) = \dfrac{\varphi^{old}(a_{N_j}) + 1}{n_i + 1}$

The new entropy on the discrete feature j when including x_j^d is:

$$H_j^{(2)}(B_i \cup \{X\}) = -\sum_{l=1}^{N_j-1} \mathsf{P}_j(a_l \in d_{ij}) \cdot \log_2 \mathsf{P}_j(a_l \in d_{ij}) - \mathsf{P}_j(x_j^d \in d_{ij}) \cdot \log_2 \mathsf{P}_j(x_j^d \in d_{ij})$$

$$= -\sum_{l=1}^{N_j-1} \frac{\varphi(a_l)}{n_i + 1} \cdot \log_2 \frac{\varphi(a_l)}{n_i + 1} - \frac{\varphi^{old}(a_{N_j}) + 1}{n_i + 1} \cdot \log_2 \frac{\varphi^{old}(a_{N_j}) + 1}{n_i + 1}$$

$$= -\sum_{l=1}^{N_j-1} \frac{\varphi(a_l)}{n_i + 1} \cdot \log_2 \frac{\varphi(a_l)}{n_i + 1} - \frac{\varphi^{old}(a_{N_j})}{n_i + 1} \cdot \log_2 \frac{\varphi^{old}(a_{N_j}) + 1}{n_i + 1} - \frac{1}{n_i + 1} \cdot \log_2 \frac{\varphi^{old}(a_{N_j}) + 1}{n_i + 1}$$

Because $1 \leq \varphi^{old}(a_{N_j})$, we have $-\log_2 \dfrac{\varphi^{old}(a_{N_j})+1}{n_i+1} < -\log_2 \dfrac{\varphi^{old}(a_{N_j})}{n_i+1}$ and $-\log_2 \dfrac{\varphi^{old}(a_{N_j})+1}{n_i+1} < -\log_2 \dfrac{1}{n_i+1}$ Hence,

$$H_j^{(2)}(B_j \cup \{X\}) < -\sum_{l=1}^{N_j-1} \frac{\varphi(a_l)}{n_i+1} \cdot \log_2 \frac{\varphi(a_l)}{n_i+1} - \frac{\varphi^{old}(a_{N_j})}{n_i+1} \cdot \log_2 \frac{\varphi^{old}(a_{N_j})}{n_i+1} - \frac{1}{n_i+1} \cdot \log_2 \frac{1}{n_i+1}$$

$$= -\sum_{l=1}^{N_j} \frac{\varphi(a_l)}{n_i+1} \cdot \log_2 \frac{\varphi(a_l)}{n_i+1} - \frac{1}{n_i+1} \cdot \log_2 \frac{1}{n_i+1} = H_j^{(1)}(B_j \cup \{X\})$$

As a result, the change in the entropy for the hyperbox B_i on the j-th discrete attribute:

$$Z_j^{(2)} = H_j^{(2)}(B_j \cup \{X\}) - H_j(B_j) < H_j^{(1)}(B_j \cup \{X\}) - H_j(B_j) = Z_j^{(1)}$$

Therefore, the change in the entropy for each discrete dimension of the hyperbox B_i will takes its maximum value if the categorical value on that categorical dimension of the input pattern does not appear in the current list of categorical values stored on that discrete dimension. This property is proved. $\qquad\square$

II. Proof of Property 2 from the Main Paper

Property 2. *The upper bound of the change in the entropy for every categorical dimension j of B_i depends on the current number of samples included in B_i. That is:*

$$Z_j \leq \log_2(n_i+1) - \frac{n_i}{n_i+1} \log_2 n_i$$

Proof. Based on the Property 1, we know that the maximum entropy change occurs on the j-th categorical dimension when it accommodates a new categorical value which does not exist in its current values. Hence, we obtain:

$$Z_j \leq Z_j^{(1)} = H_j^{(1)}(B_i \cup \{X\}) - \frac{n_i}{n_i+1} H_j(B_i)$$

$$= -\sum_{l=1}^{N_j} \frac{\varphi(a_l)}{n_i+1} \cdot \log_2 \frac{\varphi(a_l)}{n_i+1} - \frac{1}{n_i+1} \cdot \log_2 \frac{1}{n_i+1} + \frac{n_i}{n_i+1} \cdot \sum_{l=1}^{N_j} \frac{\varphi(a_l)}{n_i} \cdot \log_2 \frac{\varphi(a_l)}{n_i}$$

$$= -\frac{1}{n_i+1} \sum_{l=1}^{N_j} [\varphi(a_l) \cdot [\log_2 \varphi(a_l) - \log_2(n_i+1)]] + \frac{\log_2(n_i+1)}{n_i+1} + \frac{1}{n_i+1} \cdot \sum_{l=1}^{N_j} \varphi(a_l) \cdot [\log_2 \varphi(a_l) - \log_2 n_i]$$

$$= \frac{\log_2(n_i+1)}{n_i+1} \sum_{l=1}^{N_j} \varphi(a_l) + \frac{\log_2(n_i+1)}{n_i+1} - \frac{\log_2 n_i}{n_i+1} \cdot \sum_{l=1}^{N_j} \varphi(a_l)$$

$$= \frac{\log_2(n_i+1)}{n_i+1} \left[\sum_{l=1}^{N_j} \varphi(a_l) + 1 \right] - \frac{\log_2 n_i}{n_i+1} \cdot \sum_{l=1}^{N_j} \varphi(a_l)$$

We have: $\sum\limits_{l=1}^{N_j} \varphi(a_l) = n_i$. Hence,

$$Z_j^{(1)} = \frac{\log_2(n_i+1)}{n_i+1}(n_i+1) - \frac{\log_2 n_i}{n_i+1} \cdot n_i = \log_2(n_i+1) - \frac{n_i}{n_i+1} \cdot \log_2 n_i$$

As a result,

$$Z_j \leq \log_2(n_i+1) - \frac{n_i}{n_i+1} \cdot \log_2 n_i$$

The property is proved. $\qquad\square$

III. Proof of Property 3 from the Main Paper

Property 3. *The change of the entropy for each categorical dimension j always falls in the range of $[0, 1]$:*

$$0 \leq Z_j \leq 1; \quad \forall j \in [1, r]$$

Proof. First of all, we need to prove that $Z_j \leq 1; \ \forall j \in [1, r]$

Let $f(n) = \log_2(n+1) - \dfrac{n}{n+1} \cdot \log_2 n; \ (n \geq 1)$. We have the first order derivative of $f(n)$:

$$f'(n) = \frac{1}{(n+1)\ln 2} - \frac{\log_2 n}{(n+1)^2} - \frac{1}{(n+1)\ln 2} = -\frac{\log_2 n}{(n+1)^2}$$

Therefore, $f'(n) \leq 0$, $\forall n \geq 1$. As a result, $f(n)$ is decreasing on the interval $[1, +\infty]$. This leads to $f(n) \leq f(1) = 1$; $\forall n \geq 1$. Hence, $\log_2(n_i + 1) - \dfrac{n_i}{n_i + 1} \cdot \log_2 n_i \leq 1$; $\forall n_i \geq 1$. From the Property 2, we obtain

$$Z_j \leq \log_2(n_i + 1) - \frac{n_i}{n_i + 1} \cdot \log_2 n_i; \ \forall j \in [1, r]$$

As a result, $Z_j \leq 1$; $\forall j \in [1, r]$ is proved.

Next, we need to prove that $Z_j \geq 0$; $\forall j \in [1, r]$. From the Property 1, we can see that the change in the entropy on each discrete dimension j takes a smaller value if the included value x_j^d has already been in the current categorical values of d_{ij} in the hyperbox B_i. Hence, we only need to prove $Z_j \geq 0$ for this case. We obtain:

$$
\begin{aligned}
Z_j &= H_j^{(2)}(B_i \cup \{X\}) - \frac{n_i}{n_i + 1} \cdot H_j(B_i) \\
&= -\sum_{l=1}^{N_j} \mathsf{P}_j(a_l \in d_{ij} | B_i \cup \{X\}) \cdot \log_2 \mathsf{P}_j(a_l \in d_{ij} | B_i \cup \{X\}) + \frac{n_i}{n_i + 1} \cdot \sum_{l=1}^{N_j} \mathsf{P}_j(a_l \in d_{ij} | B_i) \cdot \log_2 \mathsf{P}_j(a_l \in d_{ij} | B_i) \\
&= -\sum_{l=1}^{N_j} \frac{\varphi(a_l | B_i \cup \{X\})}{n_i + 1} \cdot \log_2 \frac{\varphi(a_l | B_i \cup \{X\})}{n_i + 1} + \frac{n_i}{n_i + 1} \cdot \sum_{l=1}^{N_j} \frac{\varphi(a_l | B_i)}{n_i} \cdot \log_2 \frac{\varphi(a_l | B_i)}{n_i}
\end{aligned}
$$

To prove $Z_j \geq 0$, we need to prove that:

$$
\begin{aligned}
&n_i \cdot \frac{\varphi(a_l | B_i)}{n_i} \cdot \log_2 \frac{\varphi(a_l | B_i)}{n_i} \geq (n_i + 1) \cdot \frac{\varphi(a_l | B_i \cup \{X\})}{n_i + 1} \cdot \log_2 \frac{\varphi(a_l | B_i \cup \{X\})}{n_i + 1}, \forall l \in [1, N_j] \\
&\Leftrightarrow \varphi(a_l | B_i) \cdot \log_2 \frac{\varphi(a_l | B_i)}{n_i} \geq \varphi(a_l | B_i \cup \{X\}) \cdot \log_2 \frac{\varphi(a_l | B_i \cup \{X\})}{n_i + 1}, \forall l \in [1, N_j]
\end{aligned}
\tag{SE.1}
$$

Let $x = \varphi(a_l | B_i) \geq 1$, then $\varphi(a_l | B_i \cup \{X\}) = x + 1$ or $\varphi(a_l | B_i \cup \{X\}) = x$ because the number of elements of a_l on the j-th discrete feature can remain unchanged or increase by 1 when B_i includes X. In the case that $\varphi(a_l | B_i \cup \{X\}) = x$, we have:

$$(\text{SE.1}) \Leftrightarrow x \cdot \log_2 \frac{x}{n_i} \geq x \cdot \log_2 \frac{x}{n_i + 1}$$

This inequality is obviously true because $x \geq 1$ and $\dfrac{x}{n_i} \geq \dfrac{x}{n_i + 1} \Leftrightarrow \log_2 \dfrac{x}{n_i} \geq \log_2 \dfrac{x}{n_i + 1}$.

In the case that $\varphi(a_l | B_i \cup \{X\}) = x + 1$, we need to prove that:

$$x \cdot \log_2 \frac{x}{n_i} \geq (x + 1) \cdot \log_2 \frac{x + 1}{n_i + 1} \tag{SE.2}$$

Let $a_1, \ldots, a_m$ and $b_1, \ldots, b_m$ be non-negative numbers, and $a = \sum\limits_{k=1}^{m} a_k$, $b = \sum\limits_{k=1}^{m} b_k$, the log-sum inequality states that

$$\sum_{k=1}^{m} a_i \cdot \log_2 \frac{a_i}{b_i} \geq a \cdot \log_2 \frac{a}{b}$$

Based on this log-sum inequality, we obtain:

$$x \cdot \log_2 \frac{x}{n_i} + 1 \cdot \log_2 \frac{1}{1} \geq (x + 1) \cdot \log_2 \frac{x + 1}{n_i + 1}$$

We have $1 \cdot \log_2 \dfrac{1}{1} = 0$, thus the inequality (SE.2) is true.

From all above proofs, we obtain: $0 \leq Z_j \leq 1$; $\forall j \in [1, r]$. The property is proved. $\quad\square$

IV. Proof of Property 4 from the Main Paper

Property 4. *When the number of samples contained in B_i approaches infinity, the change of the entropy for every categorical dimension will be limited at 0. Formally,*

$$\lim_{n_i \to +\infty} Z_j = 0, \ \forall j \in [1, r] \tag{SE.3}$$

Proof. From the properties 2 and 3, we have:

$$0 \leq Z_j \leq \log_2(n_i + 1) - \frac{n_i}{n_i + 1} \cdot \log_2 n_i$$

To prove this property, therefore, we need to show that:

$$\lim_{n_i \to +\infty} \log_2(n_i + 1) - \frac{n_i}{n_i + 1} \cdot \log_2 n_i = 0$$

Let $f(n) = \log_2(n+1) - \dfrac{n}{n+1} \cdot \log_2 n = \dfrac{(n+1) \cdot \log_2(n+1) - n \cdot \log_2 n}{n+1}$, then $\lim_{n \to +\infty} f(n)$ is the indeterminate form $\dfrac{\infty}{\infty}$. Let's apply L'Hospital's rule:

$$\lim_{n \to +\infty} f(n) = \lim_{n \to +\infty} \frac{(n+1) \cdot \log_2(n+1) - n \cdot \log_2 n}{n+1}$$

$$= \lim_{n \to +\infty} \left[\log_2(n+1) + \frac{1}{\ln 2} - \log_2 n - \frac{1}{\ln 2} \right] = \lim_{n \to +\infty} \log_2(1 + \frac{1}{n}) = 0$$

The property is proved. $\square$

V. ADDITIONAL RESULTS

A. Datasets

Table S.I shows the detailed information about the datasets used for the experiments in this study. These datasets were used in the previous research regarding different solutions to learn from mixed-attribute data for the general fuzzy min-max neural network. These datasets show the diversity in the numbers of samples, classes, and features. All datasets are class-imbalanced.

TABLE S.I
EXPERIMENTAL DATASET

ID	Dataset	# Patterns	# Classes	# Continuous attributes	# Discrete attributes	Summary of discrete attributes
1	abalone	4177	28	7	1	1 feature: 3 values
2	australian	690	2	6	8	4 features: 2 values, 2 features: 3 values, 1 feature: 15 values, 1 feature: 8 values
3	cmc	1473	3	2	7	2 features: 4 values, 3 features: 2 values, 2 features: 4 values
4	dermatology	358	6	1	33	1 feature: 2 values, 1 feature: 3 values, 31 features: 4 values
5	flag	194	8	10	18	1 feature: 4 values, 1 feature: 6 values, 1 feature: 7 values, 1 feature: 10 values, 2 features: 8 values, 12 features: 2 values
6	german	1000	2	7	13	4 features: 4 values, 3 features: 5 values, 1 feature: 10 values, 3 features: 3 values, 2 features: 2 values
7	heart	270	2	7	6	3 features: 2 values, 1 feature: 4 values, 2 features: 3 values
8	japanese credit	653	2	6	9	4 features: 2 values, 3 features: 3 values, 1 feature: 14 values, 1 feature: 9 values
9	molecular biology	3190	3	0	60	All features: 4 values
10	nursery	12960	5	0	8	4 features: 3 values, 1 feature: 5 values, 2 features: 4 values, 1 feature: 2 values
11	post operative	87	3	1	7	5 features: 3 values, 2 features: 2 values
12	tae	151	3	1	4	1 feature: 2 values, 1 feature: 25 values, 1 feature: 26 values, 1 feature: 2 values
13	tic tac toe	958	2	0	9	All features: 3 values
14	zoo	101	7	1	15	All features: 2 values

B. Additional Experimental Results for the Main Paper

This section provides the interested readers with the detailed results for the experiments presented in the main paper.

Fig. S1 to Fig. S3 show the change in the average class balanced accuracy (CBA) scores for different values of α. These results support the contents in subsection IV.A.1 from the main paper. Fig S4 presents the change in the CBA values for different values of δ using the IOL-GFMM-v1 algorithm. Meanwhile, Fig. S5 shows similar results for the IOL-GFMM-v2 algorithm. These figures consolidate the comments in subsection IV.A.2 from the main paper. Fig S6 provides a graphical illustration for two ways of estimating the value of α shown in subsection IV.C.1 from the main paper.

Figs. from S7 to S9 show the distribution of the obtained values of α using different methods to find the suitable values for α. The best results in each row of the tables in this document are highlighted in bold. These figures support the explanation in subsection IV.C.1 from the main paper related to the effectiveness of each method of finding α.

Table S.II presents the average class balanced accuracy for different learning algorithms with mixed-attribute handling ability over 40 iterations (10 times repeated stratified 4-fold cross-validation). The number of generated hyperboxes from these algorithms is shown in Table S.III. These are the results for the experiment mentioned in subsection IV.B.1 from the main paper.

Table S.IV shows the average CBA for the proposed methods and the original improved online learning algorithm using different encoding techniques to transform the discrete features into numerical features. The ranks for these methods are presented in Table S.V. These results supplement to the claims in subsection IV.B.2 from the main paper.

Table S.VI describes the average CBA values for the different methods to find the appropriate values for α using the proposed learning algorithms. Their ranks are shown in Table S.VII. These are the detailed empirical results for the experiment presented in subsection IV.C.1 from the main paper.

Table S.VIII shows the average CBA values for four learning algorithms using the hyper-parameter tuning approach. The ranks for these results are presented in Table S.IX. These results add more details to the content in subsection IV.C.2 from the main paper.

(a) abalone (b) australian (c) cmc

(d) dermatology (e) flag (f) german

(g) heart (h) japanese credit (i) post operative

(j) tae (k) zoo

Fig. S1. Class balanced accuracy values for different values of α ($\theta = 1, \delta = 1$).

(a) abalone

(b) australian

(c) cmc

(d) dermatology

(e) flag

(f) german

(g) heart

(h) japanese credit

(i) post operative

(j) tae

(k) zoo
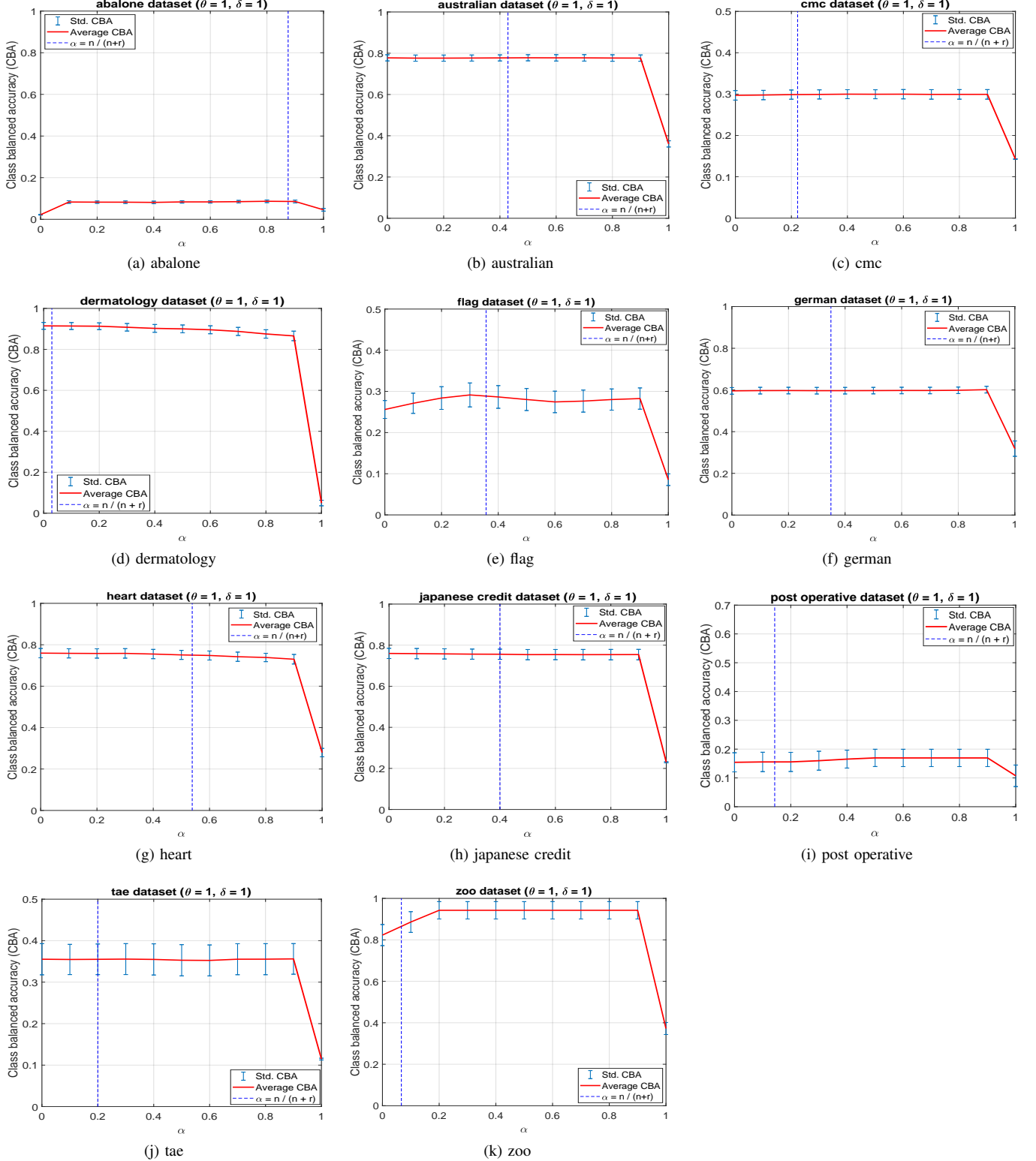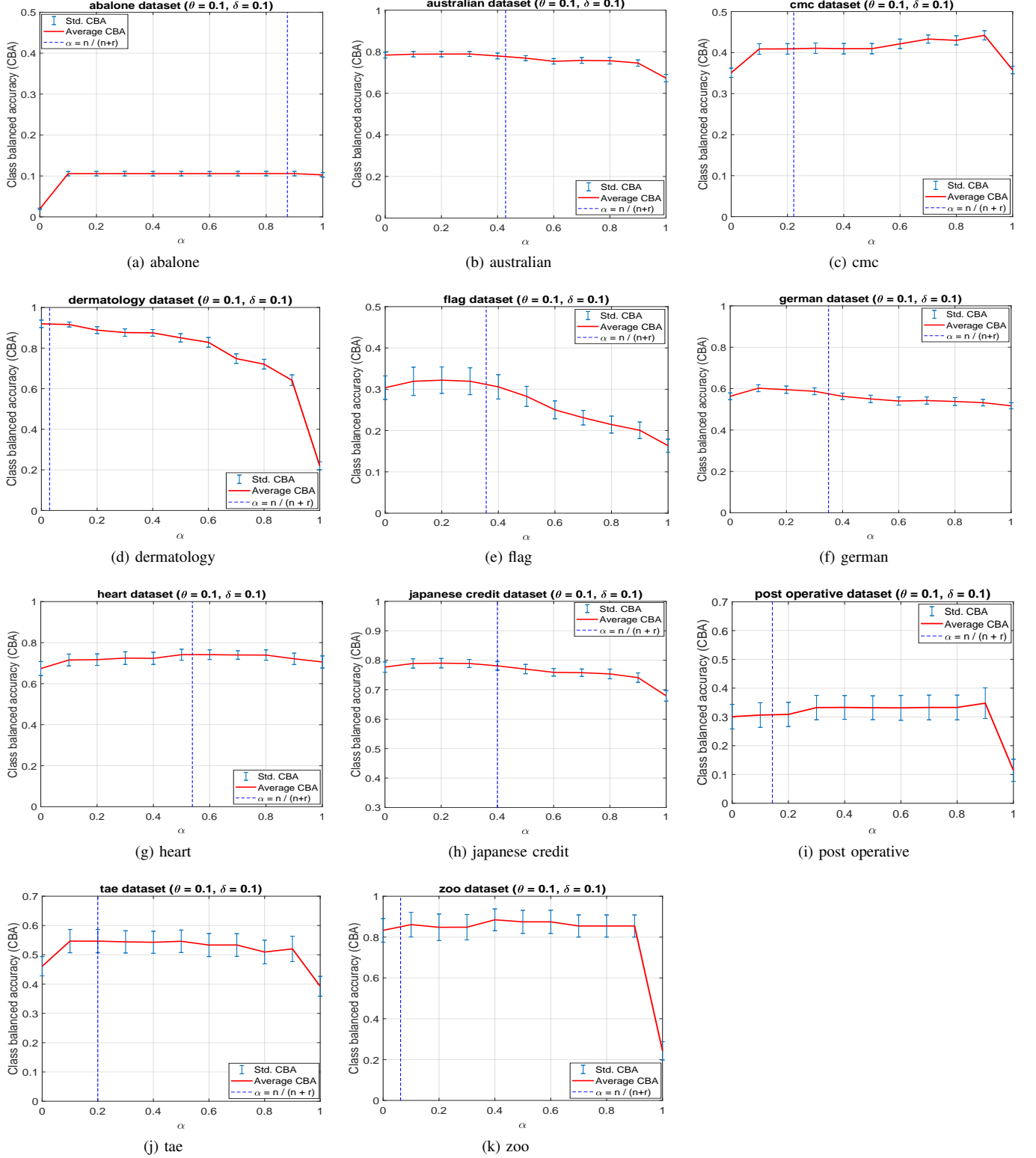
Fig. S2. Class balanced accuracy values for different values of α using the IOL-GFMM-v1 algorithm ($\theta = 0.1, \delta = 0.1$).

(a) abalone

(b) australian

(c) cmc

(d) dermatology

(e) flag

(f) german

(g) heart

(h) japanese credit

(i) post operative

(j) tae

(k) zoo
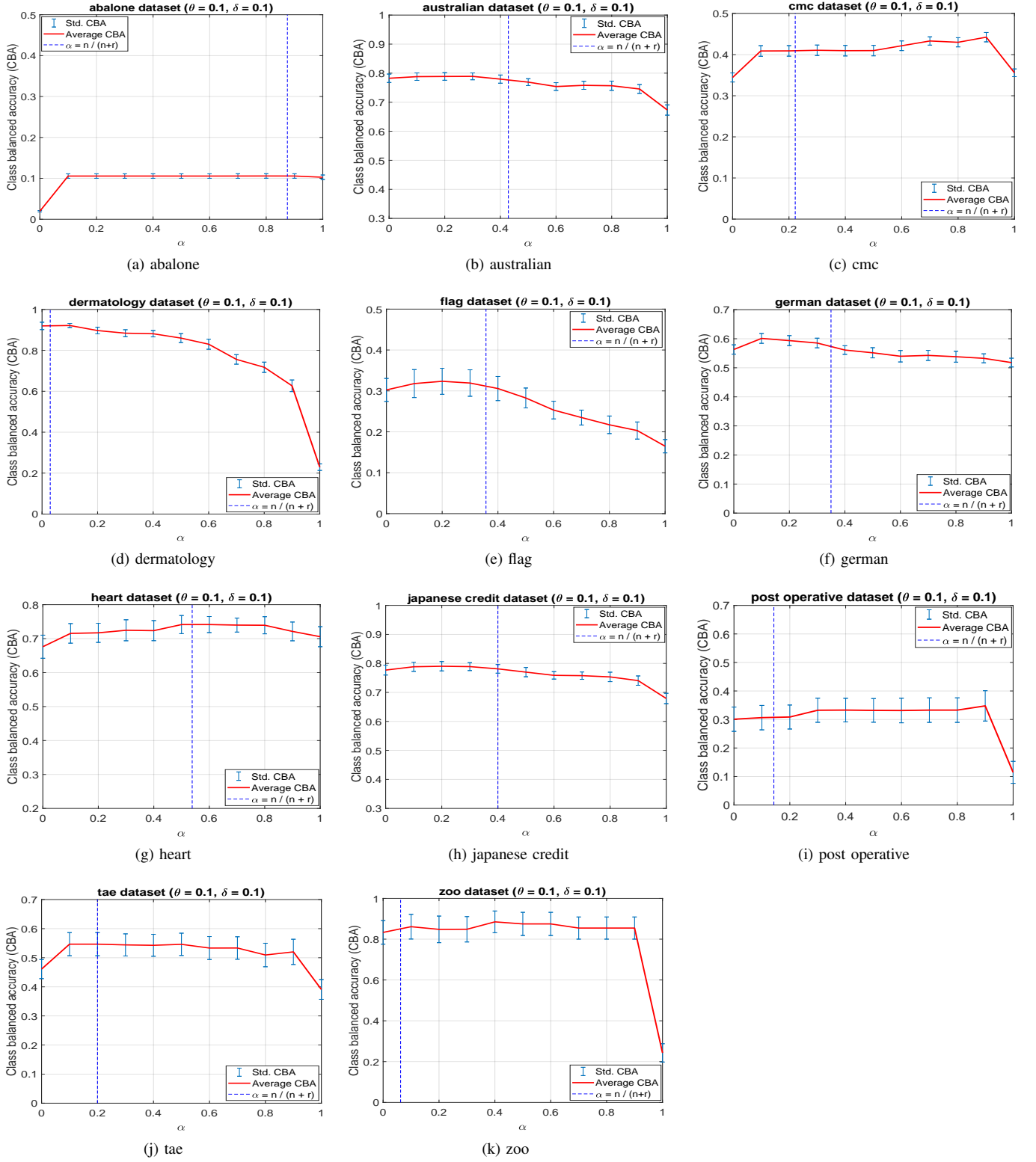
Fig. S3. Class balanced accuracy values for different values of α using the IOL-GFMM-v2 algorithm ($\theta = 0.1, \delta = 0.1$).
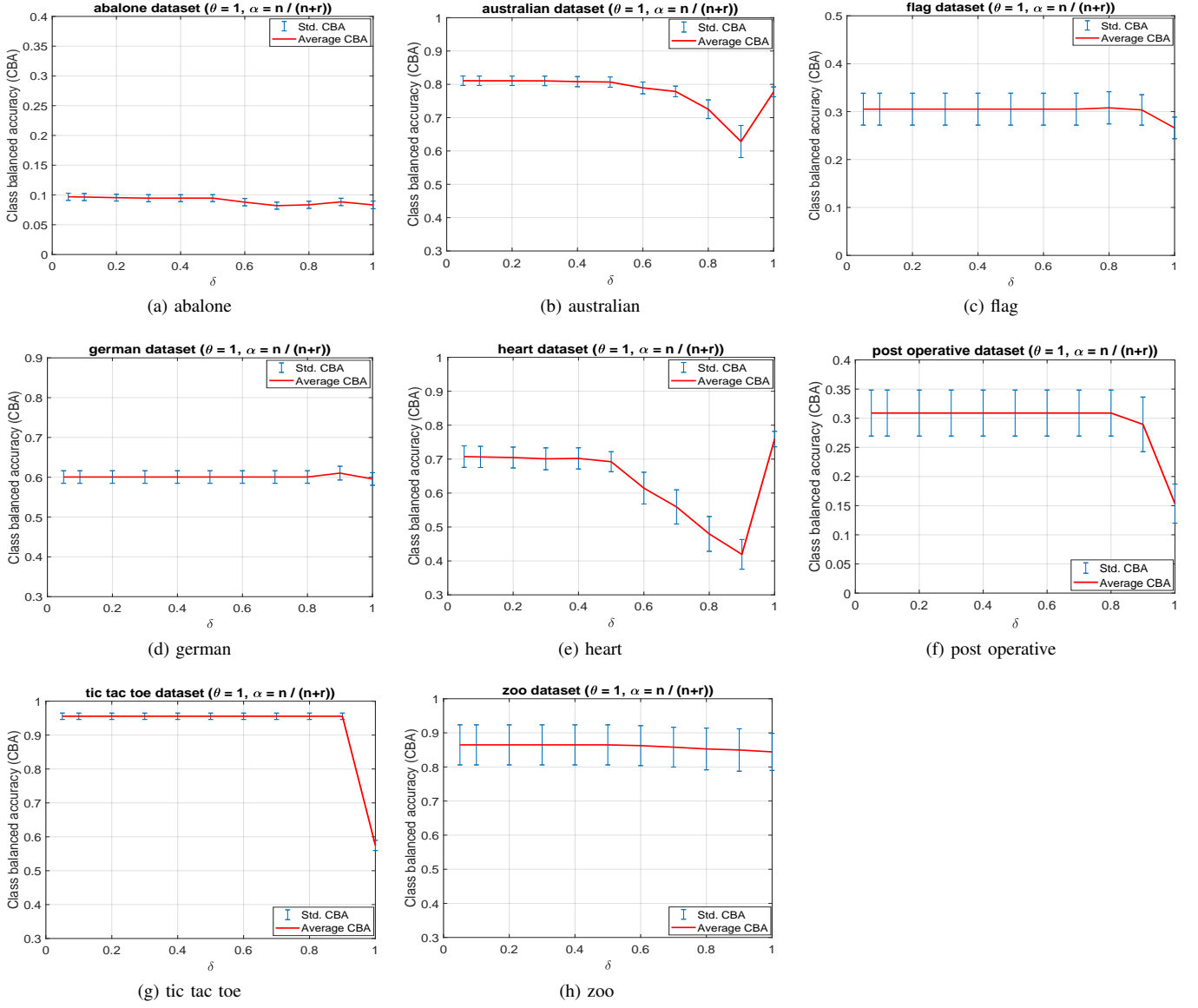
Fig. S4. Class balanced accuracy values for different values of δ using the IOL-GFMM-v1 algorithm ($\theta = 1, \alpha = n/(n + r)$).
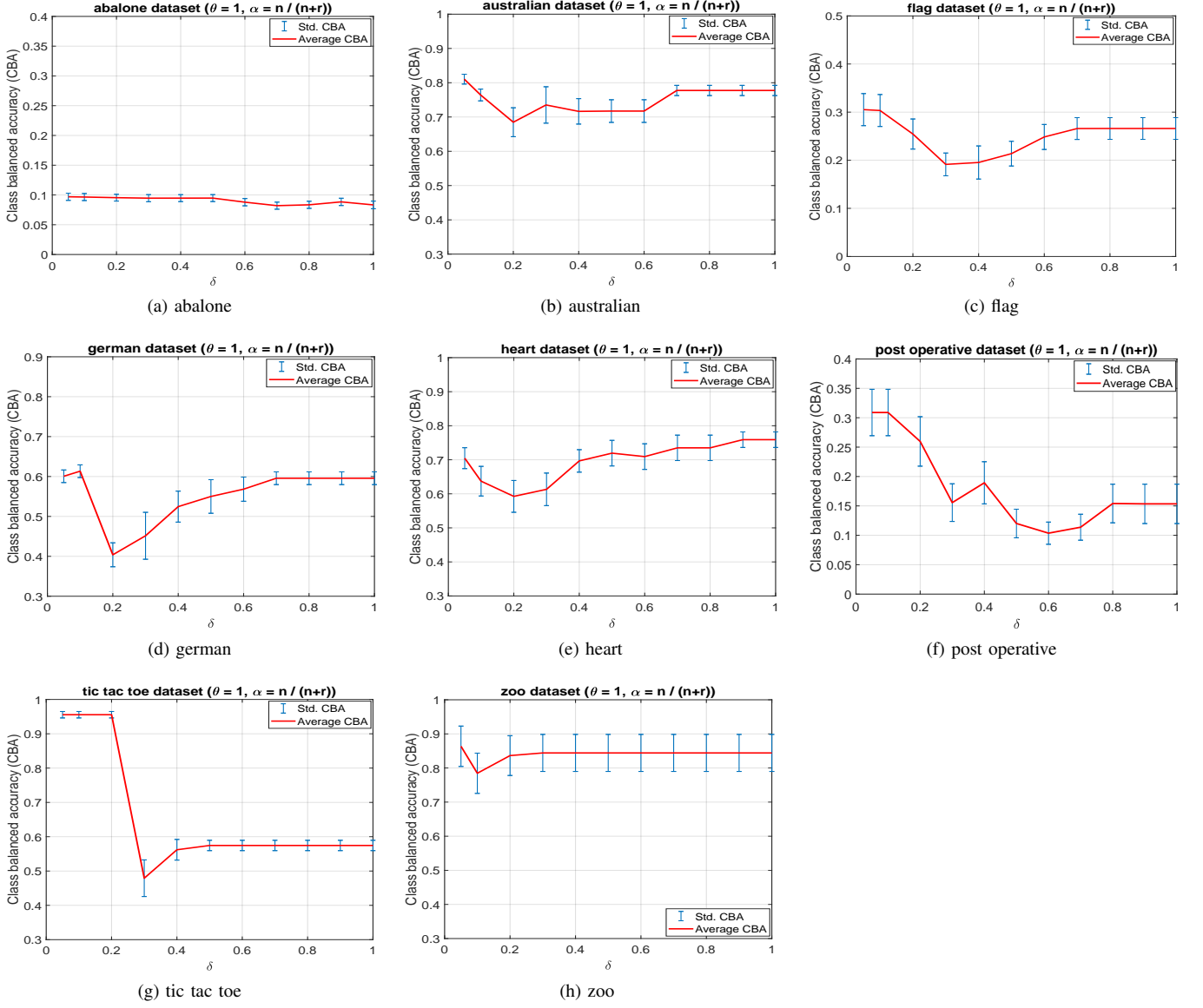
Fig. S5. Class balanced accuracy values for different values of δ using the IOL-GFMM-v2 algorithm ($\theta = 1, \alpha = n/(n + r)$).

TABLE S.II
AVERAGE CLASS BALANCED ACCURACY FOR THE PROPOSED EIOL-GFMM ALGORITHMS AND TWO EXISTING ALGORITHMS WITH THE MIXED-ATTRIBUTE LEARNING ABILITY

Dataset	θ	Onln-GFMM-M1			Onln-GFMM-M2			EIOL-GFMM-v1			EIOL-GFMM-v2		
		$\eta=0.1$	$\eta=0.7$	$\eta=1$	$\beta=0.25r$	$\beta=0.5r$	$\beta=0.75r$	$\delta=0.1$	$\delta=0.7$	$\delta=1$	$\delta=0.1$	$\delta=0.7$	$\delta=1$
abalone	0.1	0.10182	0.10156	0.1039	0.10195	0.10195	0.10195	**0.10592**	0.09893	0.0951	**0.10592**	0.09893	0.0951
	0.7	0.09182	0.08919	0.07923	0.08662	0.08662	0.08662	**0.09384**	0.07846	0.08748	**0.09384**	0.07846	0.08748
	1	0.08831	0.08357	0.0729	0.08455	0.08455	0.08455	**0.094**	0.07766	0.08545	**0.094**	0.07766	0.08545
australian	0.1	0.77345	0.76536	0.75532	0.62766	0.6276	0.6425	**0.77871**	**0.77871**	0.74137	**0.77871**	0.73931	0.74137
	0.7	0.7939	0.69939	0.62978	0.54578	0.53535	0.50845	**0.80259**	0.78758	0.77546	0.78005	0.77846	0.77546
	1	0.79382	0.69821	0.60577	0.55546	0.55456	0.50156	**0.80242**	0.78793	0.7771	0.78057	0.7771	0.7771
cmc	0.1	0.4003	0.39828	0.37507	0.39889	0.40054	**0.41933**	0.40938	0.41002	0.2987	0.40936	0.30153	0.2987
	0.7	**0.39879**	0.31526	0.28304	0.24961	0.24822	0.31226	0.36564	0.32006	0.32407	0.34705	0.33095	0.32407
	1	**0.39603**	0.31665	0.29373	0.28774	0.28759	0.19917	0.35431	0.31151	0.29882	0.32979	0.29882	0.29882
dermatology	0.1	0.89702	0.88468	0.87987	0.80026	0.80026	0.80822	0.91871	0.91871	0.81362	**0.92217**	0.81362	0.81362
	0.7	0.897	0.87034	0.86026	0.51967	0.51967	0.53411	**0.91871**	**0.91871**	0.86642	0.9095	0.86642	0.86642
	1	0.897	0.87097	0.85932	0.45787	0.45787	0.51649	**0.91871**	**0.91871**	0.91473	0.91284	0.91473	0.91473
flag	0.1	0.20758	0.2098	0.21106	0.27048	0.27417	0.27848	0.31378	0.31378	0.25518	**0.31398**	0.25518	0.25518
	0.7	0.20707	0.21187	0.22017	0.28746	0.2849	0.25168	**0.31232**	**0.31232**	0.27507	0.30981	0.27569	0.27507
	1	0.20722	0.21272	0.21607	0.25298	0.25797	0.2513	**0.31301**	**0.31301**	0.28709	0.31044	0.28691	0.28709
german	0.1	**0.59853**	0.5947	0.5906	0.49704	0.50612	0.53338	0.57323	0.57323	0.58901	0.57247	0.58943	0.58901
	0.7	**0.59975**	0.56437	0.53981	0.51698	0.51199	0.52676	0.57175	0.57175	0.57531	0.5759	0.57426	0.57531
	1	**0.60161**	0.54852	0.51234	0.37304	0.37304	0.43286	0.56632	0.56632	0.5969	0.58602	0.5969	0.5969
heart	0.1	0.72107	0.72131	0.73083	**0.7632**	0.75282	0.74892	0.74995	0.74995	0.73615	0.74995	0.73522	0.73615
	0.7	0.72325	0.71335	0.64509	0.59566	0.55006	0.54566	**0.7445**	0.73701	0.73827	0.73341	0.72268	0.73827
	1	0.72291	0.71955	0.60935	0.56718	0.53851	0.53506	0.74169	**0.76615**	0.74998	0.74213	0.74415	0.74998
japanese credit	0.1	0.77503	0.7741	0.75015	0.65456	0.64753	0.67543	0.78084	0.78084	0.70476	**0.78116**	0.70675	0.70476
	0.7	0.798	0.7099	0.60187	0.52062	0.51952	0.51664	**0.80483**	0.78195	0.75976	0.74552	0.76258	0.75976
	1	0.79766	0.7024	0.59146	0.50578	0.50578	0.50519	**0.80554**	0.78266	0.75584	0.74095	0.75584	0.75584
molecular biology	-	0.54925	0.53621	0.4805	0.43242	0.43218	0.17394	0.62653	0.63410	0.47090	**0.63435**	0.45035	0.47090
nursery	-	**0.84465**	0.38292	0.21174	0.50402	0.50402	0.50402	0.78223	0.78223	0.33203	0.78223	0.33203	0.33203
post operative	0.1	0.33633	0.31583	0.28579	0.27155	0.31368	**0.35604**	0.30645	0.30645	0.12997	0.30645	0.11659	0.12997
	0.7	**0.33491**	0.28111	0.2672	0.23059	0.23783	0.24881	0.30748	0.30748	0.1589	0.30748	0.15175	0.1589
	1	**0.33491**	0.28598	0.27953	0.24144	0.23603	0.27637	0.30888	0.30888	0.15504	0.30888	0.11313	0.15504
tae	0.1	0.54089	0.49741	0.43901	0.48868	0.52062	0.53408	**0.54668**	**0.54668**	0.43773	**0.54668**	0.39556	0.43773
	0.7	0.54286	0.42037	0.3789	0.32134	0.35273	0.45906	**0.55159**	**0.55159**	0.36061	**0.55159**	0.35133	0.36061
	1	0.53996	0.39701	0.3698	0.31717	0.3535	0.44079	**0.55181**	**0.55181**	0.35503	**0.55181**	0.31224	0.35503
tic tac toe	-	0.85823	0.8238	0.49501	0.44417	0.44417	0.44417	**0.95561**	**0.95561**	0.57448	**0.95561**	0.57448	0.57448
zoo	0.1	0.55787	0.55787	0.67455	0.86093	0.86093	0.86093	0.8647	0.86375	**0.90143**	0.80119	**0.90143**	**0.90143**
	0.7	0.6146	0.6146	0.61193	0.7827	0.7827	0.78776	0.8647	0.85801	**0.86369**	0.79869	**0.86369**	**0.86369**
	1	0.6146	0.6146	0.57089	0.7396	0.7396	0.73202	**0.8647**	0.85801	0.85744	0.79869	0.85744	0.85744

$$\alpha=\frac{CBA1+CBA3+CBA5}{CBA1+CBA2+CBA3+CBA4+CBA5+CBA6}$$

$$\alpha=\frac{(CBA1+CBA3+CBA5)\times n}{(CBA1+CBA3+CBA5)\times n+(CBA2+CBA4+CBA6)\times r}$$

n: number of numerical features
r: number of categorical features

CBA: Class Balanced Accuracy
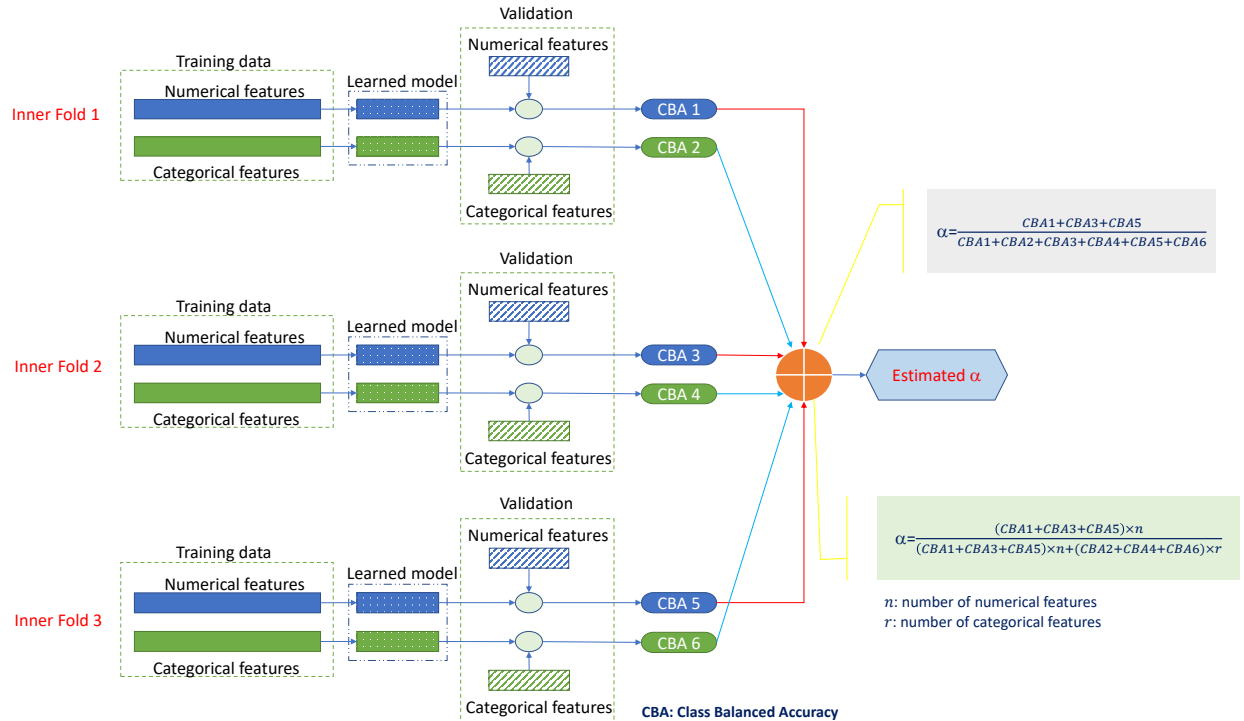
Fig. S6. A demonstration for the methods used to estimate the values for parameter α.

TABLE S.III

THE AVERAGE NUMBER OF GENERATED HYPERBOXES FROM THE PROPOSED METHOD AND OTHER EXISTING ALGORITHMS WITH MIXED-ATTRIBUTE LEARNING ABILITY

Dataset	θ	Onln-GFMM-M1			Onln-GFMM-M2			EIOL-GFMM-v1			EIOL-GFMM-v2		
		$\eta=0.1$	$\eta=0.7$	$\eta=1$	$\beta=0.25r$	$\beta=0.5r$	$\beta=0.75r$	$\delta=0.1$	$\delta=0.7$	$\delta=1$	$\delta=0.1$	$\delta=0.7$	$\delta=1$
abalone	0.1	693.95	591.725	522.825	704.15	704.15	704.15	1675.2	1318.925	919.45	1675.2	1318.925	919.45
	0.7	77.225	56.875	46.75	79.375	79.375	79.375	1665.075	605.25	934.1	1665.075	605.25	934.1
	1	63.3	45.275	37.375	65.275	65.275	65.275	1662.125	599.225	932.375	1662.125	599.225	932.375
australian	0.1	472.15	348.225	222.6	198.775	206.05	279.025	493.225	493.225	203.2	493.225	204.7	203.2
	0.7	260.275	95.825	42.7	11.675	12.875	21.875	334.45	207.5	11.85	250.55	12	11.85
	1	253.25	86.4	35.725	2	2.675	6.95	329.225	198.475	2	243.35	2	2
cmc	0.1	880.95	447.85	202.325	118.05	127.05	240	925.925	924.9	131.625	925.9	137.6	131.625
	0.7	496.325	117.2	38.675	10.85	11.775	24.475	736.475	184.975	14.25	341.1	14.5	14.25
	1	471.1	99	28.8	3	3.025	8.275	734.7	130.725	3	267.6	3	3
dermatology	0.1	268.5	212.65	105.975	38.65	38.675	71.2	268.5	268.5	46.225	255.1	46.225	46.225
	0.7	267.375	164.825	68.025	7.8	7.8	20	267.425	267.425	12.125	220.825	12.125	12.125
	1	267.375	163.85	67.1	6	6	19.05	267.425	267.425	6	220.35	6	6
flag	0.1	143.2	137.5	107.825	101.95	105.075	124.35	143.2	143.2	101.95	141.175	101.95	101.95
	0.7	140.975	123.3	44.725	23.55	25.475	49.5	140.975	140.975	23.55	132.675	23.825	23.55
	1	140.675	116.625	34.2	8	9.65	31.55	140.675	140.675	8	131.5	8.025	8
german	0.1	747.925	711.3	549.55	524.225	547.45	669.3	748.225	748.225	523.7	744.975	526.225	523.7
	0.7	726.225	299.575	115.225	32.25	36.375	87.725	738.825	738.825	32.6	666.725	32.975	32.6
	1	714.925	201.1	69.575	2	2.125	10.3	735.425	735.425	2	516	2	2
heart	0.1	198.675	197.475	181.375	181.225	183.525	188.5	199.1	199.1	181.425	199.1	182.3	181.425
	0.7	92.05	71.5	16.9	12.5	13.4	17.425	103.45	73.175	12.725	97.625	13.1	12.725
	1	74.925	53.625	7.25	2	3.375	4.85	89.85	50.8	2	76.925	2.2	2
japanese_credit	0.1	451.175	340.15	219.775	198.65	202	233.25	468.875	468.875	202.55	468.3	203.725	202.55
	0.7	252.775	95.675	42.475	12.35	12.95	15.775	320.425	210.3	12.725	197.325	12.95	12.725
	1	246.7	87.475	35.275	2	2	4.425	315.675	198.4	2	183.9	2	2
molecular_biology	-	2271.125	2034.025	1060.925	3.9	178.425	1743.325	2279.6	2279.6	3	2195.1	5.55	3
nursery	-	5434.325	156.4	33.075	5	5	5.175	9720	9720	5	9720	5	5
post_operative	0.1	58.575	38.9	13.2	7	9.05	19.1	59.25	59.25	7	59.25	8.7	7
	0.7	55.1	30.075	8.7	4.825	5.175	11.15	56.125	56.125	4.45	56.125	5.05	4.45
	1	55.05	29.375	7.475	2.75	3.95	9.425	56.05	56.05	2.75	56.05	3.625	2.75
tae	0.1	80.75	47.15	35.125	22.125	32.325	68.775	84.15	84.05	23.325	84.05	26.45	23.325
	0.7	70.55	29	21.55	5.025	6.95	37.125	77.35	77.225	6.225	77.225	6.775	6.225
	1	69.95	28.425	20.575	3.025	4.625	33.325	76.75	76.625	3	76.625	3.8	3
tic_tac_toe	-	639.125	239.275	18.175	2	2	2.05	718.5	718.5	2	718.5	2	2
zoo	0.1	48.4	48.4	13.375	13.375	13.375	14.5	49.35	45.625	13.375	26.65	13.375	13.375
	0.7	44.975	44.975	8	8	8	8.55	46	39.95	8	19.275	8	8
	1	44.975	44.975	7	7	7	7.25	46	39.95	7	18.15	7	7

TABLE S.IV

THE AVERAGE CLASS BALANCED ACCURACY FOR THE PROPOSED METHOD AND THE IOL-GFMM ALGORITHM USING ENCODING TECHNIQUES

Dataset	$\theta(=\delta)$	CatBoost	One-hot	LOO	Label	Target	James-Stein	Helmert	Sum	EIOL-GFMM-v1	EIOL-GFMM-v2
abalone	0.1	0.1077	0.11216	0.08367	**0.11217**	0.11161	0.11161	0.11216	0.11215	0.10592	0.10592
	0.7	**0.11117**	0.04521	0.04599	0.07214	0.0544	0.0544	0.04521	0.03942	0.07846	0.07846
	1	**0.11041**	0.03651	0.05247	0.07843	0.07027	0.07027	0.02491	0.02491	0.08545	0.08545
australian	0.1	0.79148	0.6283	0.70876	0.77303	0.77283	0.77321	0.62855	**0.79163**	0.77871	0.77871
	0.7	**0.81259**	0.63919	0.60169	0.79844	0.80302	0.80712	0.63943	0.80088	0.78758	0.77846
	1	0.72686	0.36155	0.45375	0.38134	0.63626	0.64024	0.36384	0.36405	**0.7771**	**0.7771**
cmc	0.1	0.27015	0.39891	0.39403	**0.4124**	0.40797	0.40849	0.39891	0.4095	0.40938	0.40936
	0.7	0.24563	0.38885	0.23629	0.37042	0.37329	0.3739	0.38885	**0.40109**	0.32006	0.33095
	1	0.28563	0.14279	0.13443	0.14279	0.14482	0.14482	0.14301	0.14301	**0.29882**	**0.29882**
dermatology	0.1	0.7371	0.05674	0.87801	0.55851	0.87721	0.85747	0.05674	0.28197	0.91871	**0.92217**
	0.7	0.75532	0.05674	0.87791	0.60397	0.88038	0.86116	0.05674	0.34796	**0.91871**	0.86642
	1	0.74119	0.75364	0.81876	0.80716	0.82864	0.81965	0.82665	0.24084	**0.91473**	**0.91473**
flag	0.1	0.2597	0.06655	0.2218	0.13868	0.14268	0.14573	0.06655	0.12138	0.31378	**0.31398**
	0.7	0.21977	0.06655	0.2205	0.13571	0.1382	0.14176	0.06655	0.12031	**0.31232**	0.27569
	1	0.12787	0.2391	0.15268	0.13549	0.17076	0.17001	0.24299	0.24084	**0.28709**	**0.28709**
german	0.1	0.57516	0.35944	0.57602	0.5219	**0.58811**	0.5854	0.35944	0.46208	0.57323	0.57247
	0.7	0.54573	0.35944	0.57281	0.51857	**0.59267**	0.59057	0.35944	0.43172	0.57175	0.57426
	1	0.3895	0.38493	0.40407	0.37334	0.36717	0.36736	0.35404	0.35404	**0.5969**	**0.5969**
heart	0.1	**0.76571**	0.6405	0.71338	0.69879	0.67343	0.69832	0.64175	0.68189	0.74995	0.74995
	0.7	**0.74487**	0.66551	0.72538	0.69216	0.69781	0.72236	0.66635	0.6776	0.73701	0.72268
	1	0.64107	0.32977	0.42858	0.47927	0.55031	0.54678	0.32641	0.32641	**0.74998**	**0.74998**
japanese credit	0.1	0.77653	0.62878	0.74217	0.77063	0.7745	0.77262	0.62878	**0.80086**	0.78084	0.78116
	0.7	**0.81413**	0.63782	0.70741	0.80831	0.80203	0.80616	0.63782	0.81319	0.78195	0.76258
	1	0.70607	0.36353	0.44061	0.49254	0.5359	0.54993	0.36467	0.36554	**0.75584**	**0.75584**
molecular biology	0.1	0.45006	0.30092	**0.68158**	0.54743	0.54642	0.51235	0.30092	0.40823	0.62742	0.63435
	0.7	0.18284	0.30092	**0.68161**	0.52217	0.54689	0.54447	0.30092	0.364	0.62742	0.47625
	1	0.38381	0.17398	0.39789	**0.47431**	0.27719	0.22743	0.17405	0.17398	0.46948	0.46948
nursery	0.1	0.34878	0.075	0.773	0.70924	0.7625	0.76726	0.075	0.45355	**0.78033**	**0.78033**
	0.7	0.34219	0.075	0.70922	0.63043	0.58198	0.58197	0.075	0.55267	**0.78033**	0.33111
	1	0.34404	0.3361	0.41684	0.42149	0.49159	**0.49174**	0.3361	0.3361	0.33111	0.33111
post operative	0.1	**0.37532**	0.31386	0.34674	0.34483	0.32674	0.32082	0.31386	0.29886	0.30645	0.30645
	0.7	**0.37125**	0.31386	0.34063	0.35297	0.33419	0.32633	0.31386	0.3535	0.30748	0.15175
	1	0.34648	0.3423	0.3623	0.36426	0.36205	**0.36685**	0.34087	0.34087	0.15504	0.15504
tae	0.1	0.27737	0.54896	0.41787	0.54959	0.53362	0.5414	0.57858	**0.58822**	0.54668	0.54668
	0.7	0.29342	0.54896	0.32724	0.40225	0.42256	0.4233	**0.57922**	0.51246	0.55159	0.35133
	1	0.29217	0.28113	0.17504	0.343	0.22983	0.28051	0.31731	0.28113	**0.35503**	**0.35503**
tic tac toe	0.1	0.40253	0.32672	0.86118	0.49864	0.84898	0.84898	0.32672	0.50738	**0.95561**	**0.95561**
	0.7	0.21604	0.32672	0.85408	0.52528	0.85278	0.85278	0.32672	0.61887	**0.95561**	0.57448
	1	0.21598	0.32672	0.34482	0.32672	0.32672	0.32672	0.32672	0.32672	**0.57448**	**0.57448**
zoo	0.1	0.73326	0.47723	**0.89851**	0.47723	0.47723	0.47723	0.47723	0.47723	0.8647	0.80119
	0.7	0.72461	0.47723	**0.89851**	0.47723	0.47723	0.47723	0.47723	0.47723	0.85801	0.86369
	1	0.66473	0.65154	**0.87506**	0.65154	0.65154	0.65154	0.65154	0.65154	0.85744	0.85744

TABLE S.V
THE RANKS FOR THE PROPOSED METHOD AND THE ORIGINAL IOL-GFMM ALGORITHM USING ENCODING TECHNIQUES

Dataset	$\theta(=\delta)$	CatBoost	One-hot	LOO	Label	Target	James-Stein	Helmert	Sum	EIOL-GFMM-v1	EIOL-GFMM-v2
abalone	0.1	7	2.5	10	**1**	5.5	5.5	2.5	4	8.5	8.5
	0.7	**1**	8.5	7	4	5.5	5.5	8.5	10	2.5	2.5
	1	**1**	8	7	4	5.5	5.5	9.5	9.5	2.5	2.5
australian	0.1	2	10	8	6	7	5	9	**1**	3.5	3.5
	0.7	**1**	9	10	5	3	2	8	4	6	7
	1	3	10	6	7	5	4	9	8	**1.5**	**1.5**
cmc	0.1	10	7.5	9	**1**	6	5	7.5	2	3	4
	0.7	9	2.5	10	6	5	4	2.5	**1**	8	7
	1	3	8.5	10	8.5	4.5	4.5	6.5	6.5	**1.5**	**1.5**
dermatology	0.1	6	9.5	3	7	4	5	9.5	8	2	**1**
	0.7	6	9.5	3	7	2	5	9.5	8	**1**	4
	1	9	8	6	7	3	5	4	10	**1.5**	**1.5**
flag	0.1	3	9.5	4	7	6	5	9.5	8	2	**1**
	0.7	4	9.5	3	7	6	5	9.5	8	**1**	2
	1	10	5	8	9	6	7	3	4	**1.5**	**1.5**
german	0.1	4	9.5	3	7	**1**	2	9.5	8	5	6
	0.7	6	9.5	4	7	**1**	2	9.5	8	5	3
	1	4	5	3	6	8	7	9.5	9.5	**1.5**	**1.5**
heart	0.1	**1**	10	4	5	8	6	9	7	2.5	2.5
	0.7	**1**	10	3	7	6	5	9	8	2	4
	1	3	8	7	6	4	5	9.5	9.5	**1.5**	**1.5**
japanese credit	0.1	4	9.5	8	7	5	6	9.5	**1**	3	2
	0.7	**1**	9.5	8	3	5	4	9.5	2	6	7
	1	3	10	7	6	5	4	9	8	**1.5**	**1.5**
molecular biology	0.1	7	9.5	**1**	4	5	6	9.5	8	3	2
	0.7	10	8.5	**1**	5	3	4	8.5	7	2	6
	1	5	9.5	4	**1**	6	7	8	9.5	2.5	2.5
nursery	0.1	8	9.5	3	6	5	4	9.5	7	**1.5**	**1.5**
	0.7	7	9.5	2	3	4	5	9.5	6	**1**	8
	1	5	7	4	3	2	**1**	7	7	9.5	9.5
post operative	0.1	**1**	6.5	2	3	4	5	6.5	10	8.5	8.5
	0.7	**1**	7.5	4	3	5	6	7.5	2	9	10
	1	5	6	3	2	4	**1**	7.5	7.5	9.5	9.5
tae	0.1	10	4	9	3	8	7	2	**1**	5.5	5.5
	0.7	10	3	9	7	6	5	**1**	4	2	8
	1	5	6.5	10	3	9	8	4	6.5	**1.5**	**1.5**
tic tac toe	0.1	8	9.5	3	7	4.5	4.5	9.5	6	**1.5**	**1.5**
	0.7	10	8.5	2	7	3.5	3.5	8.5	5	**1**	6
	1	10	6.5	3	6.5	6.5	6.5	6.5	6.5	**1.5**	**1.5**
zoo	0.1	4	7.5	**1**	7.5	7.5	7.5	7.5	7.5	2	3
	0.7	4	7.5	**1**	7.5	7.5	7.5	7.5	7.5	3	2
	1	4	7.5	**1**	7.5	7.5	7.5	7.5	7.5	2.5	2.5

TABLE S.VI

THE AVERAGE CLASS BALANCED ACCURACY FOR DIFFERENT METHODS USED TO FIND THE VALUES FOR PARAMETER α

Dataset	Algorithm	$\theta = \delta$	Tuning α	Est-α-v1	Est-α-v2	$\alpha = n/(n+r)$
abalone	EIOL-GFMM-v1	0.1	0.10452	0.10571	0.10589	0.10592
	EIOL-GFMM-v2	0.1	0.10452	0.10571	0.10589	0.10592
	Both	1	0.08795	0.08498	0.08307	0.08545
australian	EIOL-GFMM-v1	0.1	0.7806	0.77979	0.77416	0.77871
	EIOL-GFMM-v2	0.1	0.78038	0.77991	0.77406	0.77871
	Both	1	0.77654	0.77734	0.77784	0.7771
cmc	EIOL-GFMM-v1	0.1	0.43148	0.41052	0.41074	0.40938
	EIOL-GFMM-v2	0.1	0.43168	0.41074	0.41414	0.40936
	Both	1	0.29941	0.29907	0.29967	0.29882
dermatology	EIOL-GFMM-v1	0.1	0.91653	0.91871	0.88963	0.91871
	EIOL-GFMM-v2	0.1	0.91763	0.92217	0.89801	0.92217
	Both	1	0.91446	0.91473	0.91315	0.91473
flag	EIOL-GFMM-v1	0.1	0.3206	0.32098	0.30551	0.31378
	EIOL-GFMM-v2	0.1	0.32063	0.32267	0.30648	0.31398
	Both	1	0.28035	0.28494	0.28583	0.28709
german	EIOL-GFMM-v1	0.1	0.59599	0.57887	0.55533	0.57323
	EIOL-GFMM-v2	0.1	0.59563	0.5794	0.5568	0.57247
	Both	1	0.5998	0.59663	0.59677	0.5969
heart	EIOL-GFMM-v1	0.1	0.73854	0.74783	0.7465	0.74995
	EIOL-GFMM-v2	0.1	0.73854	0.7462	0.74909	0.74995
	Both	1	0.75397	0.75177	0.75383	0.74998
japanese credit	EIOL-GFMM-v1	0.1	0.76721	0.78298	0.77622	0.78084
	EIOL-GFMM-v2	0.1	0.76642	0.78392	0.77529	0.78116
	Both	1	0.75572	0.75755	0.7563	0.75584
post operative	EIOL-GFMM-v1	0.1	0.30712	0.30645	0.33384	0.30645
	EIOL-GFMM-v2	0.1	0.30712	0.30645	0.33384	0.30645
	Both	1	0.1627	0.1555	0.16801	0.15504
tae	EIOL-GFMM-v1	0.1	0.512	0.54668	0.54421	0.54668
	EIOL-GFMM-v2	0.1	0.512	0.54668	0.54421	0.54668
	Both	1	0.35715	0.35499	0.35216	0.35503
zoo	EIOL-GFMM-v1	0.1	0.84196	0.8647	0.85161	0.8647
	EIOL-GFMM-v2	0.1	0.82726	0.79839	0.85708	0.80119
	Both	1	0.91994	0.84655	0.94256	0.85744

TABLE S.VII

THE RANKS FOR DIFFERENT METHODS USED TO FIND THE VALUES FOR PARAMETER α

Dataset	Algorithm	$\theta = \delta$	Tuning α	Est-α-v1	Est-α-v2	$\alpha = n/(n+r)$
abalone	EIOL-GFMM-v1	0.1	4	3	2	1
	EIOL-GFMM-v2	0.1	4	3	2	1
	Both	1	1	3	4	2
australian	EIOL-GFMM-v1	0.1	1	2	4	3
	EIOL-GFMM-v2	0.1	1	2	4	3
	Both	1	4	2	1	3
cmc	EIOL-GFMM-v1	0.1	1	3	2	4
	EIOL-GFMM-v2	0.1	1	3	2	4
	Both	1	2	3	1	4
dermatology	EIOL-GFMM-v1	0.1	3	1.5	4	1.5
	EIOL-GFMM-v2	0.1	3	1.5	4	1.5
	Both	1	3	1.5	4	1.5
flag	EIOL-GFMM-v1	0.1	2	1	4	3
	EIOL-GFMM-v2	0.1	2	1	4	3
	Both	1	4	3	2	1
german	EIOL-GFMM-v1	0.1	1	2	4	3
	EIOL-GFMM-v2	0.1	1	2	4	3
	Both	1	1	4	3	2
heart	EIOL-GFMM-v1	0.1	4	2	3	1
	EIOL-GFMM-v2	0.1	4	3	2	1
	Both	1	1	3	2	4
japanese_credit	EIOL-GFMM-v1	0.1	4	1	3	2
	EIOL-GFMM-v2	0.1	4	1	3	2
	Both	1	4	1	2	3
post_operative	EIOL-GFMM-v1	0.1	2	3.5	1	3.5
	EIOL-GFMM-v2	0.1	2	3.5	1	3.5
	Both	1	2	3	1	4
tae	EIOL-GFMM-v1	0.1	4	1.5	3	1.5
	EIOL-GFMM-v2	0.1	4	1.5	3	1.5
	Both	1	1	3	4	2
zoo	EIOL-GFMM-v1	0.1	4	1.5	3	1.5
	EIOL-GFMM-v2	0.1	2	4	1	3
	Both	1	2	4	1	3

(a) abalone

(b) australian

(c) cmc

(d) dermatology

(e) flag

(f) german

(g) heart

(h) japanese credit

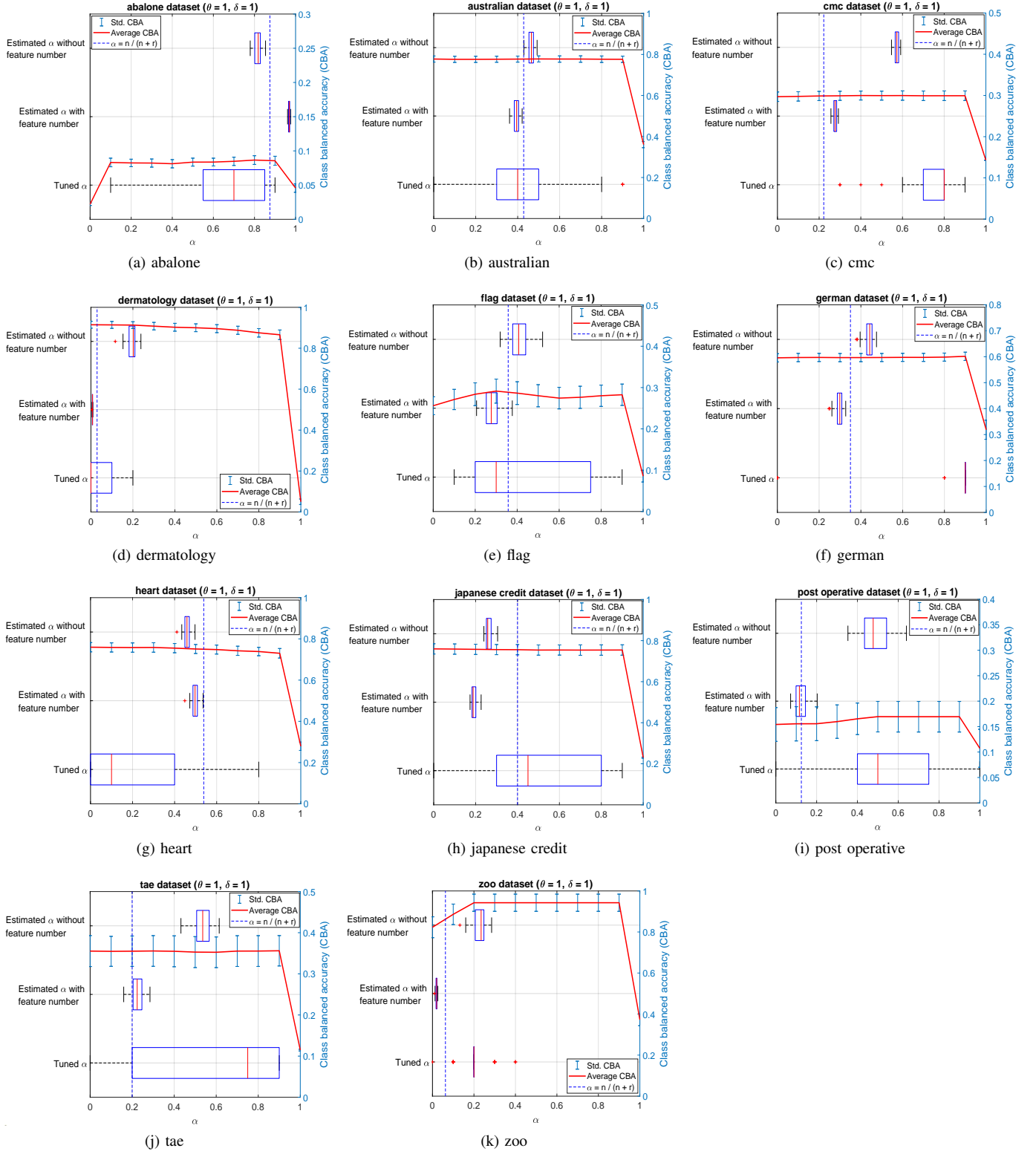(i) post operative

(j) tae

(k) zoo

Fig. S7. Class balanced accuracy values and range of obtained α for different ways of estimating α ($\theta = 1, \delta = 1$).
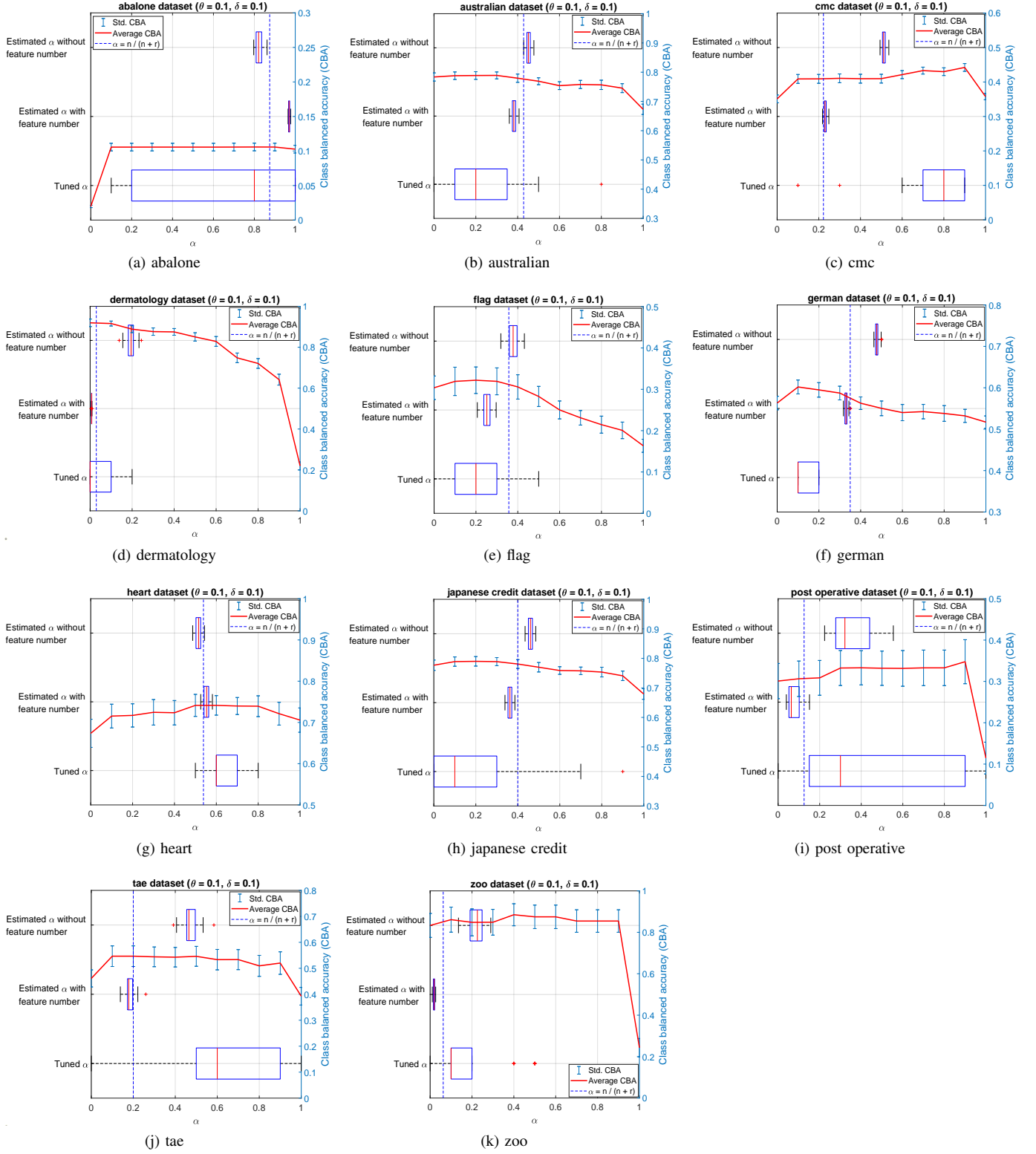
Fig. S8. Class balanced accuracy values and range of obtained α for different ways of estimating α (using EIOL-GFMM-v1 with $\theta = 0.1, \delta = 0.1$).

(a) abalone

(b) australian

(c) cmc

(d) dermatology

(e) flag

(f) german

(g) heart

(h) japanese credit

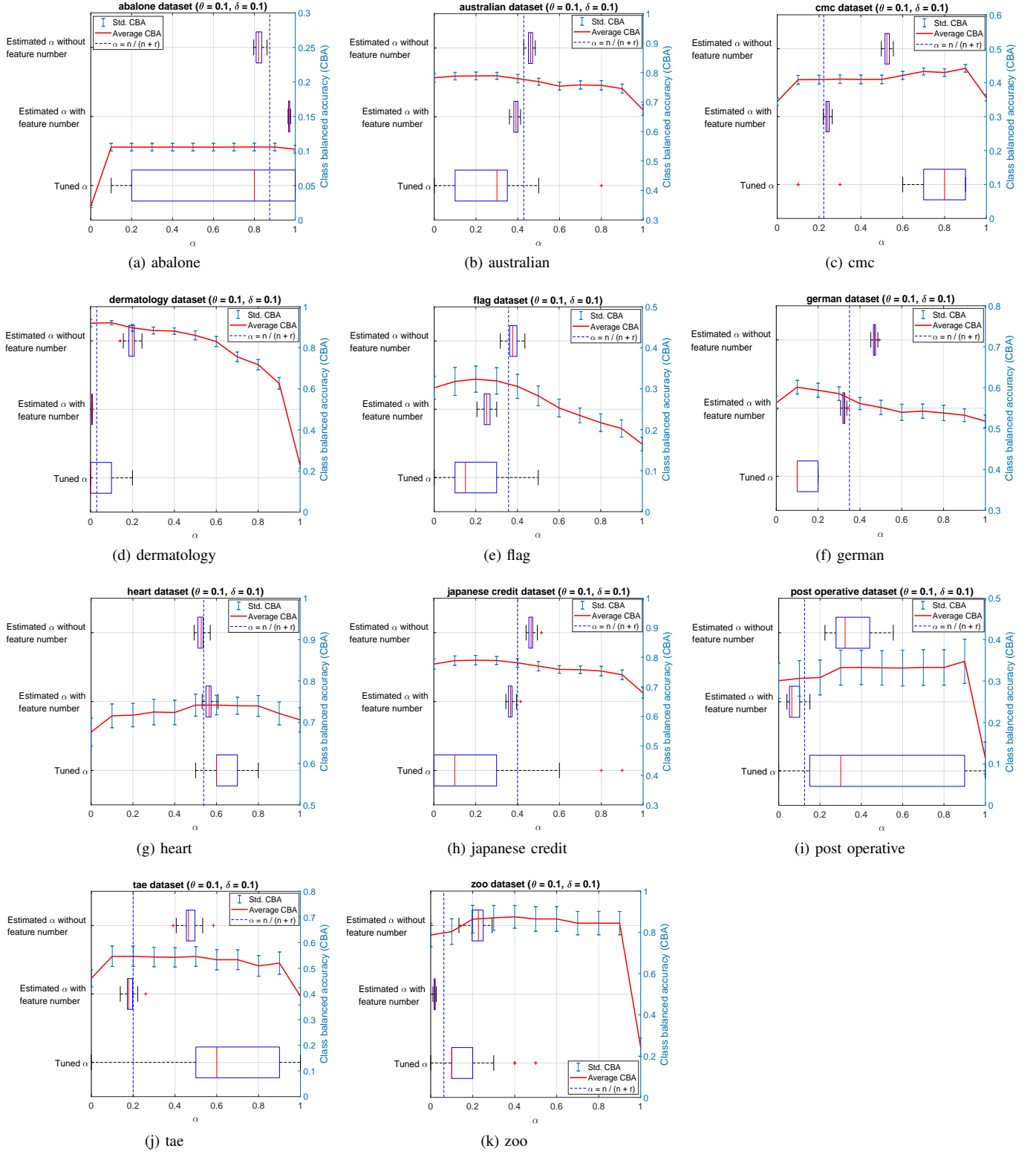(i) post operative

(j) tae

(k) zoo

Fig. S9. Class balanced accuracy values and range of obtained α for different ways of estimating α (using EIOL-GFMM-v2 with $\theta = 0.1, \delta = 0.1$).

TABLE S.VIII

THE AVERAGE CLASS BALANCED ACCURACY FOR THE LEARNING ALGORITHMS USING THE HYPER-PARAMETER TUNING METHOD

Dataset	Onln-GFMM-M1	Onln-GFMM-M2	EIOL-GFMM-v1	EIOL-GFMM-v2
abalone	0.10072	0.09932	**0.10431**	**0.10431**
australian	0.78501	0.79201	**0.79484**	0.78588
cmc	0.39265	0.40692	**0.42634**	0.42522
dermatology	0.87378	0.83522	**0.91563**	0.91497
flag	0.21735	0.27828	**0.30107**	0.28806
german	0.58233	0.55034	**0.60345**	0.59929
heart	0.72481	**0.7772**	0.76922	0.75861
japanese credit	0.763	0.76685	**0.79294**	0.79211
post operative	**0.32856**	0.29945	0.3582	0.31890
tae	**0.48682**	0.4853	0.47482	0.44618
zoo	0.67941	0.8648	**0.87179**	0.85685

TABLE S.IX

THE RANK FOR THE LEARNING ALGORITHMS USING THE HYPER-PARAMETER TUNING METHOD

Dataset	Onln-GFMM-M1	Onln-GFMM-M2	EIOL-GFMM-v1	EIOL-GFMM-v2
abalone	3	4	**1.5**	**1.5**
australian	4	2	**1**	3
cmc	4	3	**1**	2
dermatology	3	4	**1**	2
flag	4	3	**1**	2
german	3	4	**1**	2
heart	4	**1**	2	3
japanese_credit	4	3	**1**	2
post_operative	**1**	4	3	2
tae	**1**	2	3	4
zoo	4	2	**1**	3
Average	3.182	2.909	**1.5**	2.409