# An Effective Multi-Resolution Hierarchical Granular Representation based Classifier using General Fuzzy Min-Max Neural Network

Thanh Tung Khuat, Fang Chen, and Bogdan Gabrys, *Senior Member, IEEE*

***Abstract*—Motivated by the practical demands for simplification of data towards being consistent with human thinking and problem solving as well as tolerance of uncertainty, information granules are becoming important entities in data processing at different levels of data abstraction. This paper proposes a method to construct classifiers from multi-resolution hierarchical granular representations (MRHGRC) using hyperbox fuzzy sets. The proposed approach forms a series of granular inferences hierarchically through many levels of abstraction. An attractive characteristic of our classifier is that it can maintain a high accuracy in comparison to other fuzzy min-max models at a low degree of granularity based on reusing the knowledge learned from lower levels of abstraction. In addition, our approach can reduce the data size significantly as well as handle the uncertainty and incompleteness associated with data in real-world applications. The construction process of the classifier consists of two phases. The first phase is to formulate the model at the greatest level of granularity, while the later stage aims to reduce the complexity of the constructed model and deduce it from data at higher abstraction levels. Experimental analyses conducted comprehensively on both synthetic and real datasets indicated the efficiency of our method in terms of training time and predictive performance in comparison to other types of fuzzy min-max neural networks and common machine learning algorithms.**

***Index Terms*—Information granules, granular computing, hyperbox, general fuzzy min-max neural network, classification, hierarchical granular representation.**

## I. Introduction

HIERARCHICAL problem solving, where the problems are analyzed in a variety of granularity degrees, is a typical characteristic of the human brain [1]. Inspired by this ability, granular computing was introduced. One of the critical features of granular computing is to model the data as high-level abstract structures and to tackle problems based on these representations similar to structured human thinking [2]. Information granules (IGs) [3] are underlying constructs of the granular computing. They are abstract entities describing important properties of numeric data and formulating knowledge pieces from data at a higher abstraction level. They play a critical role in the concise description and abstraction of numeric data [4]. Information granules have also contributed to quantifying the limited numeric precision in data [5].

Utilizing information granules is one of the problem-solving methods based on decomposing a big problem into sub-tasks which can be solved individually. In the world of big data, one regularly departs from specific data entities and discover general rules from data via encapsulation and abstraction. The use of information granules is meaningful when tackling the five Vs of big data [6], i.e., volume, variety, velocity, veracity, and value. Granulation process gathering similar data together contributes to reducing the data size, and so the volume issue is addressed. The information from many heterogeneous sources can be granulated into various granular constructs, and then several measures and rules for uniform representation are proposed to fuse base information granules as shown in [7]. Hence, the data variety is addressed. Several studies constructed the evolving information granules to adapt to the changes in the streams of data as in [8]. The variations of information granules in a high-speed data stream assist in tackling the velocity problem of big data. The process of forming information granules is often associated with the removal of outliers and dealing with incomplete data [6]; thus the veracity of data is guaranteed. Finally, the multi-resolution hierarchical architecture of various granular levels can disregard some irrelevant features but highlight facets of interest [9]. In this way, the granular representation may help with cognitive demands and capabilities of different users.

A multi-dimensional hyperbox fuzzy set is a fundamental conceptual vehicle to represent information granules. Each fuzzy min-max hyperbox is determined by the minimum and maximum points and a fuzzy membership function. A classifier can be built from the hyperbox fuzzy sets along with an appropriate training algorithm. We can extract a rule set directly from hyperbox fuzzy sets or by using it in combination with other methods such as decision trees [10] to account for the predictive results. However, a limitation of hyperbox-based classifiers is that their accuracy at the low level of granularity (corresponding to large-sized hyperboxes) decreases. In contrast, classifiers at the high granularity level are more accurate, but the building process of classifiers at this level is time-consuming, and it is difficult to extract the rule set interpretable for predictive outcomes because of the high complexity of resulting models. Hence, it is desired to construct a simple classifier with high accuracy. In addition, we expect to observe the change in the predictive results at

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.

F. Chen (email: Fang.Chen@uts.edu.au) is with Data Science Centre, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia.

different data abstraction levels. This paper proposes a method of constructing a high-precision classifier at the high data abstraction level based on the knowledge learned from lower abstraction levels. On the basis of classification errors on the validation set, we can predict the change in the accuracy of the constructed classifier on unseen data, and we can select an abstraction level satisfying both acceptable accuracy and simple architecture on the resulting classifier. Furthermore, our method is likely to expand for large-sized datasets due to the capability of parallel execution during the constructing process of core hyperboxes at the highest level of granularity. In our method, the algorithm starts with a relatively small value of maximum hyperbox size ($\theta$) to produce base hyperbox fuzzy sets, and then this threshold is increased in succeeding levels of abstraction whose inputs are the hyperbox fuzzy sets formed in the previous step. By using many hierarchical resolutions of granularity, the information captured in earlier steps is transferred to the classifier at the next level. Therefore, the classification accuracy is still maintained at an acceptable value when the resolution of training data is low.

Data generated from complex real-world applications frequently change over time, so the machine learning models used to predict behaviors of such systems need the efficient online learning capability. Many studies considered the online learning capability when building machine learning models such as [11], [12], [13], [14], [15], [16], and [17]. Fuzzy min-max neural networks proposed by Simpson [11] and many of its improved variants only work on the input data in the form of points. In practice, due to the uncertainty and some abnormal behaviors in the systems, the input data include not only crisp points but also intervals. To address this problem, Gabrys and Bargiela [12] introduced a general fuzzy min-max (GFMM) neural network, which can handle both fuzzy and crisp input samples. By using hyperbox fuzzy sets for the input layer, this model can accept the input patterns in the granular form and process data at a high-level abstract structure. As a result, our proposed method used a similar mechanism as in the general fuzzy min-max neural network to build a series of classifiers through different resolutions, where the small-sized resulting hyperbox fuzzy sets generated in the previous step become the input to be handled at a higher level of abstraction (corresponding to a higher value of the allowable hyperbox size). Going through different resolution degrees, the valuable information in the input data is fuzzified and reduced in size, but our method helps to preserve the amount of knowledge contained in the original datasets. This capability is illustrated via the slow decline in the classification accuracy. In some cases, the predictive accuracy increases at higher levels of abstraction because the noise existing in the detailed levels is eliminated.

Building on the principles of developing GFMM classifiers with good generalization performance discussed in [18], this paper employs different hierarchical representations of granular data with various hyperbox sizes to select a compact classifier with acceptable accuracy at a high level of abstraction. Hierarchical granular representations using consecutive maximum hyperbox sizes form a set of multi-resolution hyperbox-based models, which can be used to balance the trade-off between efficiency and simplicity of the classifiers. A model with high resolution corresponds to the use of a small value of maximum hyperbox size, and vice versa. A choice of suitable resolution level results in better predictive accuracy of the generated model. Our main contributions in this paper can be summarized as follows:

- We propose a new data classification model based on the multi-resolution of granular data representations in combination with the online learning ability of the general fuzzy min-max neural network.
- The proposed method is capable of reusing the learned knowledge from the highest granularity level to construct new classifiers at higher abstraction levels with the low trade-off between the simplification and accuracy.
- The efficiency and running time of the general fuzzy min-max classifier are significantly enhanced in the proposed algorithm.
- Our classifier can perform on large-sized datasets because of the parallel execution ability.
- Comprehensive experiments are conducted on synthetic and real datasets to prove the effectiveness of the proposed method compared to other approaches and baselines.

The rest of this paper is organized as follows. Section II presents existing studies related to information granules as well as briefly describes the online learning version of the general fuzzy min-max neural network. Section III shows our proposed method to construct a classifier based on data granulation. Experimental configuration and results are presented in Section IV. Section V concludes the main findings and discusses some open directions for future works.

## II. Preliminaries

### A. Related Work

There are many approaches to representing information granules [19]. Several typical methods include intervals [20], fuzzy sets [21], shadowed sets [22], and rough sets [23]. Our study only focuses on fuzzy sets and intervals. Therefore, related works only mention the granular representation using these two methods.

The existing studies on the granular data representation have deployed a specific clustering technique to find representative prototypes, and then build information granules from these prototypes and optimize the constructed granular descriptors. The principle of justifiable granularity [24] has been usually utilized to optimize the construction of information granules from available experimental evidence. This principle aims to make a good balance between coverage and specificity properties of the resulting granule concerning available data. The coverage property relates to how much data is located inside the constructed information granule, whereas the specificity of a granule is quantified by its length of the interval such that the shorter the interval, the better the specificity. Pedrycz and Homenda [24] made a compromise between these two characteristics by finding the parameters to maximize the product of the coverage and specificity.

Instead of just stopping at the numeric prototypes, partition matrices, or dendrograms for data clustering, Pedrycz and Bargiela [25] offered a concept of granular prototypes to capture more details of the structure of data to be clustered. Granular prototypes were formed around the resulting numeric prototypes of clustering algorithms by using some degree of granularity. Information granularity is allocated to each numeric prototype to maximize the quality of the granulation-degranulation process of generated granules. This process was also built as an optimization problem steered by the coverage criteria, i.e., maximization of the original number of data included in the information granules after degranulation.

In [5], Pedrycz developed an idea of granular models derived from the establishment of optimal allocation of information granules. The authors gave motivation and plausible explanations in bringing the numeric models to the next abstraction levels to form granular models. In the realization flow of the general principles, Pedrycz et al. [4] introduced a holistic process of constructing information granules through a two-phase procedure in a general framework. The first phase focuses on formulating numeric prototypes using fuzzy C-means, and the second phase refines each resulting prototype to form a corresponding information granule by employing the principle of justifiable granularity.

When the problem becomes complicated, one regularly splits it into smaller sub-tasks and deal with each sub-task on a single level of granularity. These actions give rise to the appearance of multi-granularity computing, which aims to tackle problems from many levels of different IGs rather than just one optimal granular layer. Wang et al. [1] conducted a review on previous studies of granular computing and claimed that multi-granularity joint problem resolving is a valuable research direction to enhance the quality and efficiency of solutions based on using multiple levels of information granules rather than only one granular level. This is the primary motivation for our study to build suitable classifiers from many resolutions of granular data representations.

All the above methods of building information granules are based on the clustering techniques and affected by a pre-determined parameter, i.e., the number of clusters. The resulting information granules are only summarization of the original data at a higher abstraction level, and they did not use the class information in the constructing process of granules. The authors have not used the resulting granules to deal with classification problems either. Our work is different from these studies because we propose a method to build classifiers from various abstraction levels of data using the hyperbox fuzzy sets while maintaining the reasonable stability of classification results. In addition, our method can learn useful information from data through an online approach and the continuous adjustment of the existing structure of the model.

In the case of formulating models in a non-stationary environment, it is essential to endow them with some mechanisms to deal with the dynamic environment. In [26], Sahel et al. assessed two adaptive methods to tackle data drifting problems, i.e., retraining models using evolving data and deploying incremental learning algorithms. Although these approaches improved the accuracy of classifiers compared to non-adaptive learners, the authors indicated a great demand on building robust techniques with high reliability for dynamic operating environments. To meet the continuous changing in data and the adaptation of the analytic system to this phenomenon, Peters and Weber [27] suggested a framework, namely Dynamic Clustering Cube, to classify dynamic granular clustering methods. Al-Hmouz et al. [8] introduced evolvable data models through the dynamic changing of temporal or spatial IGs. The number of information granules formed from prototypes of data can increase or decrease through merging or splitting existing granules according to the varying complexity of data streams. In addtion to the ability to merge existing hyperboxes for the construction of granular hierarchies of hyperboxes, our proposed method also has the online learning ability, so it can be used to tackle classification problems in a dynamic environment.

Although this study is built based on the principles of GFMM classifiers, it differs from the GFMM neural network with adaptive hyperbox sizes [12]. In [12], the classifier was formed at the high abstraction level with large-sized hyperboxes and then repeating many times the process of traversing entire training dataset to build additional hyperboxes at lower abstraction levels with the aim of covering patterns missed by large-sized hyperboxes due to the contraction procedure. This operation can make the final classifier complex with a large number of hyperboxes at different levels of granularity coexisting in a single classifier, and overfitting phenomenon on the training set is more likely to happen. In contrast, our method begins with the construction process of the classifier at the highest resolution of training patterns with small-sized hyperboxes to capture detailed information and relationships among data points located in the vicinity of each other. After that, at higher levels of abstraction, we do not use the data points from the training set. Instead, we reuse the hyperboxes generated from the preceding step. For each input hyperbox, the membership value with the hyperboxes in the current step are computed, and if the highest membership degree is larger than a pre-defined threshold, the aggregation process is performed to form hyperboxes with higher abstraction degrees. Our research is also different from the approach presented in [28], where the incremental algorithm was employed to reduce the data complexity by creating small-sized hyperboxes, and then these hyperboxes were used as training inputs of an agglomerative learning algorithm with a higher abstraction level. The method in [28] only constructs the classifier with two abstraction levels, while our algorithm can generate a series of classifiers at hierarchical resolutions of abstraction levels. In addition, the agglomerative learning in [28] is time-consuming, especially in large-sized datasets. Therefore, when the number of generated hyperboxes using the incremental learning algorithm on large-sized training sets is large, the agglomerative learning algorithm takes a long time to formulate hyperboxes. On the contrary, our method takes advantage of the incremental learning ability to build rapidly classifiers through different levels of the hierarchical resolutions.

### B. General Fuzzy Min-Max Neural Network

General fuzzy min-max (GFMM) neural network [12] is a generalization and extension of the fuzzy min-max neural network (FMNN) [11]. It combines both classification and clustering in a unified framework and can deal with both fuzzy and crisp input samples. The architecture of the general fuzzy min-max neural network comprises three layers, i.e., an input layer, a hyperbox fuzzy set layer, and an output (class) layer, shown in Fig. 1.

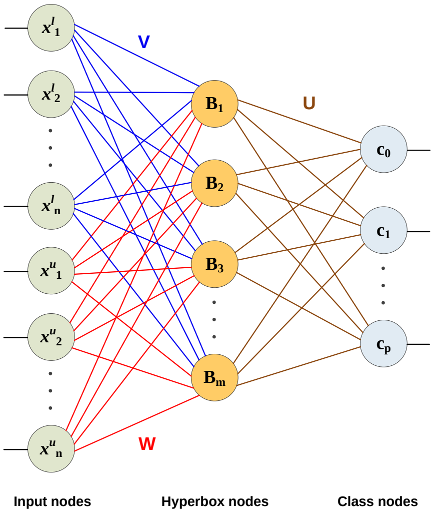


Fig. 1. The architecture of GFMM neural network.

The input layer contains $2 \cdot n$ nodes, where $n$ is the number of dimensions of the problem, to fit with the input sample $X = [X^l, X^u]$, determined within the n-dimensional unit cube $\mathbf{I}^n$. The first $n$ nodes in the input layer are connected to $m$ nodes of the second layer, which contains hyperbox fuzzy sets, by the minimum points matrix $\mathbf{V}$. The remaining $n$ nodes are linked to $m$ nodes of the second layer by the maximum points matrix $\mathbf{W}$. The values of two matrices $\mathbf{V}$ and $\mathbf{W}$ are adjusted during the learning process. Each hyperbox $B_i$ is defined by an ordered set: $B_i = \{X, V_i, W_i, b_i(X, V_i, W_i)\}$, where $V_i \subset \mathbf{V}, W_i \subset \mathbf{W}$ are minimum and maximum points of hyperbox $B_i$ respectively, $b_i$ is the membership value of hyperbox $B_i$ in the second layer, and it is also the transfer function between input nodes in the first layer and hyperbox nodes in the second layer. The membership function $b_i$ is computed using (1) [12].

$$b_i(X, V_i, W_i) = \min_{j=1\dots n} (\min([1 - f(x^u_j - w_{ij}, \gamma_j)], \\ [1 - f(v_{ij} - x^l_j, \gamma_j)])) \tag{1}$$

where $f(r, \gamma) = \begin{cases} 1, & \text{if } r\gamma > 1 \\ r\gamma, & \text{if } 0 \le r\gamma \le 1 \\ 0, & \text{if } r\gamma < 0 \end{cases}$ is the threshold function and $\gamma = [\gamma_1, \dots, \gamma_n]$ is a sensitivity parameter regulating the speed of decrease of the membership values.

Hyperboxes in the middle layer are fully connected to the third-layer nodes by a binary valued matrix $\mathbf{U}$. The elements in the matrix $\mathbf{U}$ are computed as follows:

$$u_{ij} = \begin{cases} 1, & \text{if hyperbox } B_i \text{ represents class } c_j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $B_i$ is the hyperbox of the second layer, and $c_j$ is the $j^{th}$ node in the third layer. Output of each node $c_j$ in the third layer is a membership degree to which the input pattern $X$ fits within the class $j$. The transfer function of each node $c_j$ in $p+1$ nodes belonging to the third layer is computed as:

$$c_j = \max_{i=1}^{m} b_i \cdot u_{ij} \tag{3}$$

Node $c_0$ is connected to all unlabeled hyperboxes of the middle layer. The values of nodes in the output layer can be fuzzy if they are computed directly from (3) or crisp in the case that the node with the largest value is assigned to 1 and the others get a value of zero [12].

The incremental learning algorithm for the GFMM neural network to adjust the values of two matrices $\mathbf{V}$ and $\mathbf{W}$ includes four steps, i.e., initialization, expansion, hyperbox overlap test, and contraction, in which the last three steps are repeated. In the initialization stage, each hyperbox $B_i$ is initialized with $V_i = 1$ and $W_i = 0$. For each input sample $X$, the algorithm finds the hyperbox $B_i$ with the highest membership value representing the same class as $X$ to verify two expansion conditions, i.e., maximum allowable hyperbox size and class label compatibility as shown in the supplemental file. If both criteria are met, the selected hyperbox is expanded. If no hyperbox meets the expansion conditions, a new hyperbox is created to accommodate the input data. Otherwise, if the hyperbox $B_i$ is expanded in the prior step, it will be checked for an overlap with other hyperboxes $B_k$ as follows. If the class of $B_i$ is equal to zero, then the hyperbox $B_i$ must be checked for overlap with all existing hyperboxes; otherwise, the overlap checking is only performed between $B_i$ and hyperboxes $B_k$ representing other classes. If the overlap occurs, a contraction process is carried out to remove the overlapping area by adjusting the sizes of hyperboxes according to the dimension with the smallest overlapping value. Four cases of the overlap checking and contraction procedures were presented in detail in the supplemental file and can be found in [12].

## III. Proposed Methods

### A. Overview

The learning process of the proposed method consists of two phases. The first phase is to rapidly construct small-sized hyperbox fuzzy sets from similar input data points. This phase is performed in parallel on training data segments. The data in each fragment can be organized according to two modes. The first way is called heterogeneous mode, which uses the data order read from the input file. The second mode is homogeneous, in which the data are sorted according to groups; each group contains data from the same class. The main purpose of the second phase is to decrease the complexity of the model by reconstructing phase-1 hyperboxes with a higher abstraction level.

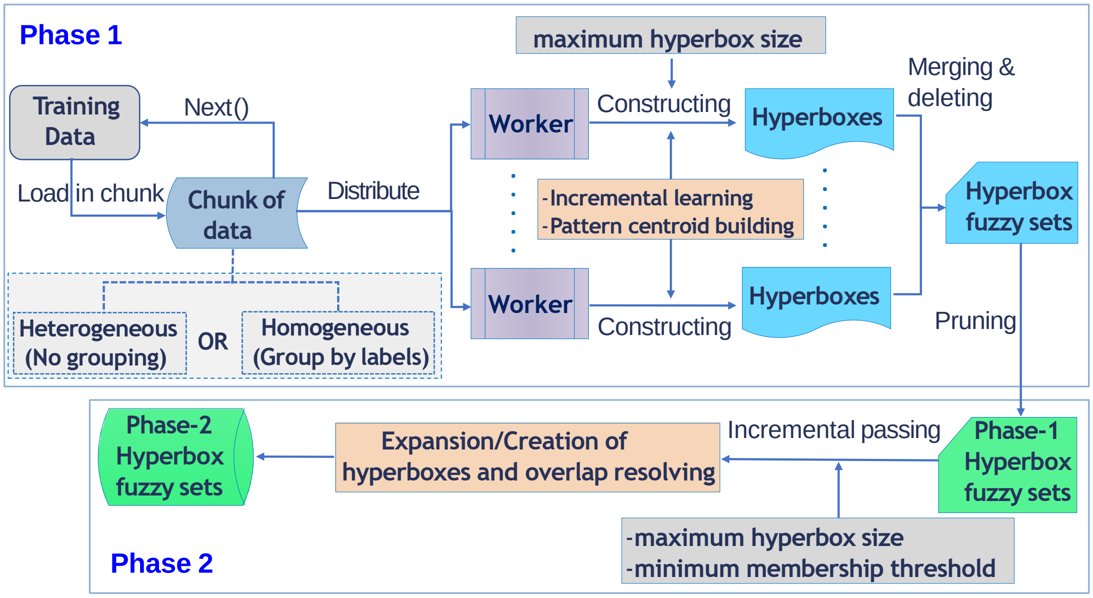


Fig. 2. Pipeline of the training process of the proposed method.

In the first step of the training process, the input samples are split into disjoint sets and are then distributed to different computational workers. On each worker, we build an independent general fuzzy min-max neural network. When all training samples are handled, all created hyperboxes at different workers are merged to form a single model. Hyperboxes completely included in other hyperboxes representing the same class are eliminated to reduce the redundancy and complexity of the final model. After combining hyperboxes, the pruning procedure needs to be applied to eliminate noise and low-quality hyperboxes. The resulting hyperboxes are called phase-1 hyperboxes.

However, phase-1 hyperboxes have small sizes, so the complexity of the system can be high. As a result, all these hyperboxes are put through phase-2 of the granulation process with a gradual increase in the maximum hyperbox sizes. At a larger value of the maximum hyperbox size, hyperboxes at a low level of abstraction will be reconstructed with a higher data abstraction degree. Previously generated hyperboxes are fetched one at a time, and they are aggregated with newly constructed hyperboxes at the current granular representation level based on a similarity threshold of the membership degree. This process is repeated for each input value of the maximum hyperbox sizes. The whole process of the proposed method is shown in Fig. 2. Based on the classification error of resulting classifiers on the validation set, one can select an appropriate predictor satisfying both simplicity and precision. The following part provides the core concepts for both phases of our proposed method in a form of mathematical descriptions. The details of Algorithm 1 for the phase 1 and Algorithm 2 corresponding to the phase 2 as well as their implementation aspects are shown in the supplemental material. The readers can refer to this document to find more about the free text descriptions, pseudo-codes, and implementation pipeline of the algorithms.

## B. Formal Description

Consider a training set of $N_{Tr}$ data vectors, $\mathbf{X}^{(Tr)} = \{X_i^{(Tr)} : X_i^{(Tr)} \in \mathbb{R}^n, i = 1, \ldots, N_{Tr}\}$, and the corresponding classes, $\mathbf{C}^{(Tr)} = \{c_i^{(Tr)} : c_i^{(Tr)} \in \mathbb{N}, i = 1, \ldots, N_{Tr}\}$; a validation set of $N_V$ data vectors, $\mathbf{X}^{(V)} = \{X_i^{(V)} : X_i^{(V)} \in \mathbb{R}^n, i = 1, \ldots, N_V\}$, and the corresponding classes, $\mathbf{C}^{(V)} = \{c_i^{(V)} : c_i^{(V)} \in \mathbb{N}, i = 1, \ldots, N_V\}$. The details of the method are formally described as follows.

### 1) Phase 1:

Let $n_w$ be the number of workers to execute the hyperbox construction process in parallel. Let $\mathbb{F}_j(\mathbf{X}_j^{(Tr)}, \mathbf{C}_j^{(Tr)}, \theta_0)$ be the procedure to construct hyperboxes on the $j^{th}$ worker with maximum hyperbox size $\theta_0$ using training data $\{\mathbf{X}_j^{(Tr)}, \mathbf{C}_j^{(Tr)}\}$. Procedure $\mathbb{F}_j$ is a modified fuzzy min-max neural network model which only creates new hyperboxes or expands existing hyperboxes. It accepts the overlapping regions among hyperboxes representing different classes, because we expect to capture rapidly similar samples and group them into specific clusters by small-sized hyperboxes without spending much time on computationally expensive hyperbox overlap test and resolving steps. Instead, each hyperbox $B_i$ is added a centroid $G_i$ of patterns contained in that hyperbox and a counter $N_i$ to store the number of data samples covered by it in addition to maximum and minimum points. This information is used to classify data points located in the overlapping regions. When a new pattern $X$ is presented to the classifier, the operation of building the pattern centroid for each hyperbox (line 12 and line 15 in Algorithm 1) is performed according to (4).

$$G_i^{new} = \frac{N_i \cdot G_i^{old} + X}{N_i + 1} \tag{4}$$

where $G_i$ is the sample centroid of the hyperbox $B_i$, $N_i$ is the number of current samples included in the $B_i$. Next, the number of samples is updated: $N_i = N_i + 1$. It is noted that

$G_i$ is the same as the first pattern covered by the hyperbox $B_i$ when $B_i$ is newly created.

After the process of building hyperboxes in all workers finishes, merging step is conducted (lines 19-23 in Algorithm 1), and it is mathematically represented as:

$$\mathbf{M} = \{B_i | B_i \in \bigcup_{j=1}^{n_w} \mathbb{F}_j(\mathbf{X}_j^{(Tr)}, \mathbf{C}_j^{(Tr)}, \theta_0)\} \tag{5}$$

where $\mathbf{M}$ is the model after performing the merging procedure. It is noted that hyperboxes contained in the larger hyperboxes representing the same class are eliminated (line 24 in Algorithm 1) and the centroids of larger hyperboxes are updated using (6).

$$G_1^{new} = \frac{N_1 \cdot G_1^{old} + N_2 \cdot G_2^{old}}{N_1 + N_2} \tag{6}$$

where $G_1$ and $N_1$ are the centroid and the number of samples of the larger sized hyperbox, $G_2$ and $N_2$ are the centroid and the number of samples of the smaller sized hyperbox. The number of samples in the larger sized hyperbox is also updated: $N_1 = N_1 + N_2$. This whole process is similar to the construction of an ensemble classifier at the model level shown in [29].

Pruning step is performed after merging hyperboxes to remove noise and low-quality hyperboxes (line 26 in Algorithm 1). Mathematically, it is defined as:

$$\mathbf{H}_0 = \begin{cases} \mathbf{M}_1 = \mathbf{M} \setminus \{B_k | \mathcal{A}_k < \alpha \ \vee \mathcal{A}_k = Nil\}, \text{ if } \mathcal{E}_V(\mathbf{M}_1) \leq \mathcal{E}_V(\mathbf{M}_2) \\ \mathbf{M}_2 = \mathbf{M} \setminus \{B_k | \mathcal{A}_k < \alpha\}, \text{ otherwise} \end{cases} \tag{7}$$

where $\mathbf{H}_0$ is the final model of stage 1 after applying the pruning operation, $\mathcal{E}_V(\mathbf{M}_i)$ is the classification error of the model $\mathbf{M}_i$ on the validation set $\{\mathbf{X}^{(V)}, \mathbf{C}^{(V)}\}$, $\alpha$ is the minimum accuracy of each hyperbox to be retained, and $\mathcal{A}_k$ is the predictive accuracy of hyperbox $B_k \in \mathbf{M}$ on the validation set defined as follows:

$$\mathcal{A}_k = \frac{\sum\limits_{j=1}^{S_k} \mathcal{R}_{kj}}{\sum\limits_{j=1}^{S_k} (\mathcal{R}_{kj} + \mathcal{I}_{kj})} \tag{8}$$

where $S_k$ is the number of validation samples classified by hyperbox $B_k$, $\mathcal{R}_k$ is the number of samples predicted correctly by $B_k$, and $\mathcal{I}_k$ is the number of incorrect predicted samples. If $S_k = 0$, then $A_k = Nil$.

The classification step of unseen samples using model $\mathbf{H}_0$ is performed in the same way as in the GFMM neural network with an exception of the case of many winning hyperboxes with the same maximum membership value. In such a case, we compute the Euclidean distance from the input sample $X$ to centroids $G_i$ of winning hyperboxes $B_i$ using (9). If the input sample is a hyperbox, $X$ is the coordinate of the center point of that hyperbox.

$$d(X, G_i) = \sqrt{\sum_{j=1}^{n} (x_j - G_{ij})^2} \tag{9}$$

The input pattern is then classified to the hyperbox $B_i$ with the minimum value of $d(X, G_i)$.

### 2) Phase 2:

Unlike phase 1, the input data in this phase are hyperboxes generated in the previous step. The purpose of stage 2 is to reduce the complexity of the model by aggregating hyperboxes created at a higher resolution level of granular data representations. At the high level of data abstraction, the confusion among hyperboxes representing different classes needs to be removed. Therefore, the overlapping regions formed in phase 1 have to be resolved, and there is no overlap allowed in this phase. Phase 2 can be mathematically represented as:

$$\mathbf{H}_H(\Theta, m_s) = \{\mathbf{H}_i | \mathbf{H}_i = \mathbb{G}(\mathbf{H}_{i-1}, \theta_i, m_s)\}, \forall i \in [1, |\Theta|], \theta_i \in \Theta \tag{10}$$

where $\mathbf{H}_H$ is a list of models $\mathbf{H}_i$ constructed through different levels of granularity represented by maximum hyperbox sizes $\theta_i$, $\Theta$ is a list of maximum hyperbox sizes, $|\Theta|$ is the cardinality of $\Theta$, $m_s$ is the minimum membership degree of two aggregated hyperboxes, and $\mathbb{G}$ is a procedure to construct the models in phase 2 (it uses the model at previous step as input), $\mathbf{H}_0$ is the model in phase 1. The aggregation rule of hyperboxes, $\mathbb{G}$, is described as follows:

For each input hyperbox $B_h$ in $\mathbf{H}_{i-1}$, the membership values between $B_h$ and all existing hyperboxes with the same class as $B_h$ in $\mathbf{H}_i$ are computed. We select the winner hyperbox with maximum membership degree with respect to $B_h$, denoted $B_k$, to aggregate with $B_h$. The following constraints are verified before conducting the aggregation:

- Maximum hyperbox size:

$$\max(w_{hj}, w_{kj}) - \min(v_{hj}, w_{kj}) \leq \theta_i, \quad \forall j \in [1, n] \tag{11}$$

- The minimum membership degree:

$$b(B_h, B_k) \geq m_s \tag{12}$$

- Overlap test. New hyperbox aggregated from $B_h$ and $B_k$ does not overlap with any existing hyperboxes in $\mathbf{H}_i$ belonging to other classes

If hyperbox $B_k$ has not met all of the above conditions, the hyperbox with the next highest membership value is selected and the process is repeated until the aggregation step occurs or no hyperbox candidate is left. If the input hyperbox cannot be mergered with existing hyperboxes in $\mathbf{H}_i$, it will be directly inserted into the current list of hyperboxes in $\mathbf{H}_i$. After that, the overlap test operation between the newly inserted hyperbox and hyperboxes in $\mathbf{H}_i$ representing other classes is performed, and then the contraction process will be executed to resolve overlapping regions. The algorithm is iterated for all input hyperboxes in $\mathbf{H}_{i-1}$.

The classification process for unseen patterns using the hyperboxes in phase 2 is realized as in the GFMM neural network. A detailed decription of the implementation steps for the proposed method can be found in the supplemental material.

## C. Missing Value Handling

The proposed method can deal with missing values since it inherits this characteristic from the general fuzzy min-max neural network as shown in [30]. A missing feature

$x_j$ is assumed to be able to receive values from the whole range, and it is presented by a real-valued interval as follows: $x_j^l = 1, x_j^u = 0$. By this initialization, the membership value associated with the missing value will be one, and thus the missing value does not cause any influence on the membership function of the hyperbox. During the training process, only observed features are employed for the update of hyperbox minimum and maximum vertices while missing variables are disregarded automatically. For the overlap test procedure in phase 2, only the hyperboxes which satisfy $v_{ij} \leq w_{ij}$ for all dimensions $j \in [1, n]$ are verified for the undesired overlapping areas. The second change is related to the way of calculating the membership value for the process of hyperbox selection or classification step of an input sample with missing values. Some hyperboxes' dimensions have not been set, so the membership function shown in (1) is changed to $b_i(X, \min(V_i, W_i), \max(W_i, V_i))$. With the use of this method, the training data uncertainty is represented in the classifier model.

## IV. Experiments

Marcia et al. [31] argued that data set selection poses a considerable impact on conclusions of the accuracy of learners, and then the authors advocated for considering properties of the datasets in experiments. They indicated the importance of employing artificial data sets constructed based on previously defined characteristics. In these experiments, therefore, we considered two types of synthetic datasets with linear and non-linear class boundaries. We also changed the number of features, the number of samples, and the number of classes for synthetic datasets to assess the variability in the performance of the proposed method. In practical applications, the data are usually not ideal as well as not following a standard distribution rule and including noisy data. Therefore, we also carried out experiments on real datasets with diversity in the numbers of samples, features, and classes.

For medium-sized real datasets such as *Letter*, *Magic*, *White wine quality*, and *Default of credit card clients*, the density-preserving sampling (DPS) method [32] was used to separate the original datasets into training, validation, and test sets. For large-sized datasets, we used the hold-out method for splitting datasets, which is the simplest and the least computationally expensive approach to assessing the performance of classifiers because more advanced resampling approaches are not essential for large amounts of data [32]. The classification model is then trained on the training dataset. The validation set is used for the pruning step and evaluating the performance of the constructed classifier aiming to select a suitable model. The testing set is employed to assess the efficiency of the model on unseen data.

The experiments aim to answer the following questions:

- How is the classification accuracy of the predictor using multi-resolution hierarchical granular representations improved in comparison to the model using a fixed granulation value?
- How good is the performance of the proposed method compared with other types of fuzzy min-max neural networks and popular algorithms based on other data representations such as support vector machines, Naive Bayes, and decision tree?
- Whether we can obtain a classifier with high accuracy at high abstraction levels of granular representations?
- Whether we can rely on the performance of the model on validation sets to select a good model for unseen data, which satisfies both simplicity and accuracy?
- How good is the ability of handling missing values in datasets without requiring data imputation?
- How critical are the roles of the pruning process and the use of sample centroids?

The limitation of runtime for each execution is seven days. If an execution does not finish within seven days, it will be terminated, and the result is reported as N/A. In the experiments, we set up parameters as follows: $n_w = 4, \alpha = 0.5, m_s = 0.4, \gamma = 1$ because they gave the best results on a set of preliminary tests with validation sets for the parameter selection. All datasets are normalized to the range of [0, 1] because of the characteristic of the fuzzy min-max neural networks. Most of the datasets except the *SUSY* datasets utilized the value of 0.1 for $\theta_0$ in phase 1, and $\Theta = \{0.2, 0.3, 0.4, 0.5, 0.6\}$ for different levels of granularity in phase 2. For the *SUSY* dataset, due to the complexity and limitation of runtime for the proposed method and other compared types of fuzzy min-max neural networks, the $\theta_0 = 0.3$ was used for phase 1, and $\Theta = \{0.4, 0.5, 0.6\}$ was employed for phase 2. For Naive Bayes, we used Gaussian Naive Bayes (GNB) algorithm for classification. The radial basis function (RBF) was used as a kernel function for the support vector machines (SVM). We used the default setting parameters in the *scikit-learn* library for Gaussian Naive Bayes, SVM, and decision tree (DT) in the experiments. The performance of the proposed method was also compared to other types of fuzzy min-max neural networks such as the original fuzzy min-max neural network (FMNN) [11], the enhanced fuzzy min-max neural network (EFMNN) [33], the enhanced fuzzy min-max neural network with a K-nearest hyperbox expansion rule (KNEFMNN) [34], and the general fuzzy min-max neural network (GFMMNN) [12]. These types of fuzzy min-max neural networks used the same pruning procedure as our proposed method.

Synthetic datasets in our experiments were generated by using Gaussian distribution functions, so Gaussian Naive Bayes and SVM with RBF kernel which use Gaussian distribution assumptions to classify data will achieve nearly optimal error rates because they match perfectly with underlying data distribution. Meanwhile, fuzzy min-max classifiers employ the hyperboxes to cover the input data, thus they are not an optimal representation for underlying data. Therefore, the accuracy of hyperbox-based classifiers on synthetic datasets cannot outperform the predictive accuracy of Gaussian NB or SVM with RBF kernel. However, Gaussian NB is a linear classifier, and thus, it only outputs highly accurate predictive results for datasets with linear decision boundary. In contrast, decision tree, fuzzy min-max neural networks, and SVM with RBF kernel are universal approximators, and they can deal effectively with both linear and non-linear classification problems.

All experiments were conducted on the computer with Xeon 6150 2.7GHz CPU and 180GB RAM. We repeated each experiment five times to compute the average training time. The accuracy of types of fuzzy min-max neural networks remains the same through different iterations because they only depend on the data presentation order and we kept the same order of training samples during the experiments.

### *A. Performance of the Proposed Method on Synthetic Datasets*

The first experiment was conducted on the synthetic datasets with the linear or non-linear boundary between classes. For each synthetic dataset, we generated a testing set containing 100,000 samples and a validation set with 10,000 instances using the same probability density function as the training sets.

*1) Linear Boundary Datasets:*

***Increase the number of samples:***

We kept both the number of dimensions $n = 2$ and the number of classes $C = 2$ the same, and only the number of samples was changed to evaluate the impact of the number of patterns on the performance of classifiers. We used Gaussian distribution to construct synthetic datasets as described in [35]. The means of the Gaussians of two classes are given as follows: $\mu_1 = [0, 0]^T, \mu_2 = [2.56, 0]^T$, and the covariance matrices are as follows:

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

With the use of these configuration parameters, training sets with different sizes (10K, 1M, and 5M samples) were formed. Fukunaga [35] indicated that the general Bayes error of the datasets formed from these settings is around 10%. We generated an equal number of samples for each class to remove the impact of imbalanced class property on the performance of classifiers. Fig. 3 shows the change in the error rates of different fuzzy min-max classifiers on the testing synthetic linear boundary datasets with the different numbers of training patterns when the level of granularity ($\theta$) changes. The other fuzzy min-max neural networks used the fixed value of $\theta$ to construct the model, while our method builds the model starting from the defined lowest value of $\theta$ (phase 1) to the current threshold. For example, the model at $\theta = 0.3$ in our proposed method is constructed with $\theta = 0.1$, $\theta = 0.2$, and $\theta = 0.3$.

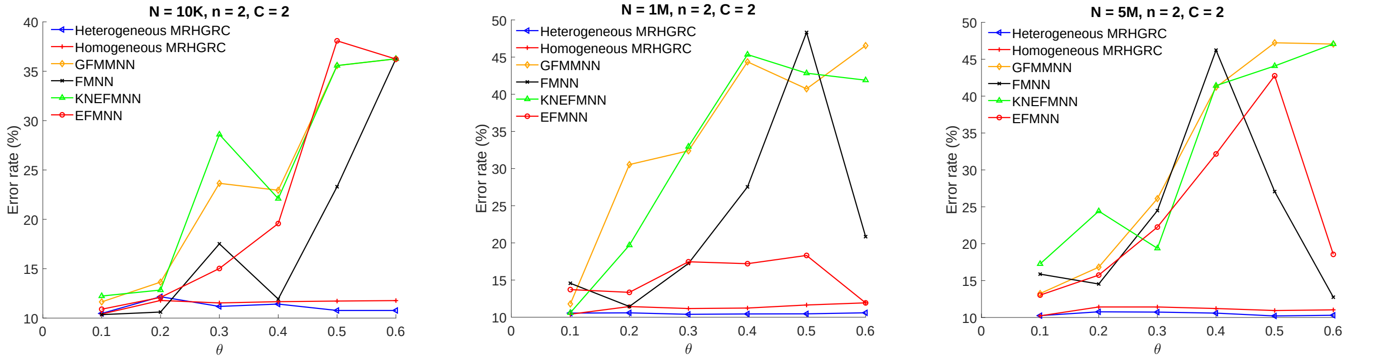
It can be seen from Fig. 3 that the error rates of our method are lower than those of other fuzzy min-max classifiers, especially in high abstraction levels of granular representations. At high levels of abstraction (corresponding to high values of $\theta$), the error rates of other classification models are relatively high, while our proposed classifier still maintains the low error rate, just a little higher than the error at a high-resolution level of granular data. The lowest error rates of the different classifiers on validation ($E_V$) and testing ($E_T$) sets, as well as total training time for six levels of abstraction, are shown in Table I. Best results are highlighted in bold in each table. The training time reported in this paper consists of time for reading training files and model construction.

TABLE I
THE LOWEST ERROR RATES AND TRAINING TIME OF CLASSIFIERS ON SYNTHETIC LINEAR BOUNDARY DATASETS WITH DIFFERENT NUMBER OF SAMPLES ($n = 2, C = 2$)

| **N** | **Algorithm** | $\min E_V$ | $\min E_T$ | $\theta_V$ | $\theta_T$ | **Time (s)** |
|---|---|---|---|---|---|---|
| 10K | He-MRHGRC | 10.25 | 10.467 | 0.1 | 0.1 | 1.1378 |
| | Ho-MRHGRC | 10.1 | 10.413 | 0.1 | 0.1 | 1.3215 |
| | GFMM | 11.54 | 11.639 | 0.1 | 0.1 | 8.6718 |
| | FMNN | 10.05 | 10.349 | 0.1 | 0.1 | 46.4789 |
| | KNEFMNN | 12.07 | 12.232 | 0.1 | 0.1 | 9.4459 |
| | EFMNN | 10.44 | 10.897 | 0.1 | 0.1 | 48.9892 |
| | GNB | **9.85** | **9.964** | - | - | 0.5218 |
| | SVM | 9.91 | 9.983 | - | - | 1.5468 |
| | DT | 15.33 | 14.861 | - | - | 0.5405 |
| 1M | He-MRHGRC | 10.31 | 10.386 | 0.3 | 0.3 | 20.0677 |
| | Ho-MRHGRC | 10.24 | 10.401 | 0.1 | 0.1 | 16.0169 |
| | GFMM | 11.47 | 11.783 | 0.1 | 0.1 | 405.4642 |
| | FMNN | 10.98 | 11.439 | 0.2 | 0.2 | 13163.1404 |
| | KNEFMNN | 10.36 | 10.594 | 0.1 | 0.1 | 413.8296 |
| | EFMNN | 11.61 | 11.923 | 0.6 | 0.6 | 10845.1280 |
| | GNB | 9.87 | **9.972** | - | - | 5.0133 |
| | SVM | **9.86** | 9.978 | - | - | 21798.2803 |
| | DT | 14.873 | 14.682 | - | - | 9.9318 |
| 5M | He-MRHGRC | 10.11 | 10.208 | 0.5 | 0.5 | 101.9312 |
| | Ho-MRHGRC | 10.04 | 10.222 | 0.1 | 0.1 | 75.2254 |
| | GFMM | 13.14 | 13.243 | 0.1 | 0.1 | 1949.2138 |
| | FMNN | 12.68 | 12.751 | 0.6 | 0.6 | 92004.7253 |
| | KNEFMNN | 17.31 | 17.267 | 0.1 | 0.1 | 1402.1173 |
| | EFMNN | 12.89 | 13.032 | 0.1 | 0.1 | 41888.5296 |
| | GNB | **9.88** | **9.976** | - | - | 22.9343 |
| | SVM | N/A | N/A | - | - | N/A |
| | DT | 15.253 | 14.692 | - | - | 70.2041 |

We can see that the accuracy of our method on unseen data using the heterogeneous data distribution (He-MRHGRC) regularly outperforms the accuracy of the classifier built based on the homogeneous data distribution (Ho-MRHGRC) using large-sized training sets. It is also observed that our method is less affected by overfitting when increasing the number of training samples while keeping the same testing set. For other types of fuzzy min-max neural networks, their error rates frequently increase with the increase in training size because of overfitting. The total training time of our algorithm is faster than that of other types of fuzzy min-max classifiers since our proposed method executes the hyperbox building process at the lowest value of $\theta$ in parallel, and we accept the overlapping areas among hyperboxes representing different classes to rapidly capture the characteristics of sample points locating near each other. The hyperbox overlap resolving step is only performed at higher abstraction levels with a smaller number of input hyperboxes.

Our proposed method also achieves better classification accuracy compared to the decision tree, but it cannot overcome the support vector machines and Gaussian Naive Bayes methods on synthetic linear boundary datasets. However, the training time of the support vector machines on large-sized datasets is costly, even becomes unacceptable on training sets with millions of patterns. The synthetic datasets were constructed based on the Gaussian distribution, so the Gaussian Naive Bayes method can reach the minimum error rate, but our approach can also obtain the error rates relatively near these optimal error values. We can observe that the best performance of the He-MRHGRC attains at quite high abstraction levels of granular representations because some noisy hyperboxes at high levels of granularity are eliminated at lower granulation levels. These results demonstrate the efficiency and scalability



(a) 10K samples

(b) 1M samples

(c) 5M samples

Fig. 3. The error rate of classifiers on synthetic linear boundary datasets with the different number of samples.

TABLE II
THE LOWEST ERROR RATES AND TRAINING TIME OF CLASSIFIERS ON SYNTHETIC LINEAR BOUNDARY DATASETS WITH DIFFERENT CLASSES ($N = 10K, n = 2$)

| C | Algorithm | min $E_V$ | min $E_T$ | $\theta_V$ | $\theta_T$ | Time (s) |
|---|---|---|---|---|---|---|
| 2 | He-MRHGRC | 10.25 | 10.467 | 0.1 | 0.1 | 1.1378 |
| | Ho-MRHGRC | 10.10 | 10.413 | 0.1 | 0.1 | 1.3215 |
| | GFMM | 11.54 | 11.639 | 0.1 | 0.1 | 8.6718 |
| | FMNN | 10.05 | 10.349 | 0.1 | 0.1 | 46.4789 |
| | KNEFMNN | 12.07 | 12.232 | 0.1 | 0.1 | 9.4459 |
| | EFMNN | 10.44 | 10.897 | 0.1 | 0.1 | 48.9892 |
| | GNB | **9.85** | **9.964** | - | - | 0.5218 |
| | SVM | 9.91 | 9.983 | - | - | 1.5468 |
| | DT | 15.33 | 14.861 | - | - | 0.5405 |
| 4 | He-MRHGRC | 19.76 | 19.884 | 0.4 | 0.4 | 1.0754 |
| | Ho-MRHGRC | 19.97 | 21.135 | 0.1 | 0.1 | 1.5231 |
| | GFMM | 22.34 | 22.515 | 0.1 | 0.1 | 10.8844 |
| | FMNN | 20.00 | 20.350 | 0.1 | 0.1 | 65.7884 |
| | KNEFMNN | 20.54 | 20.258 | 0.1 | 0.1 | 12.5618 |
| | EFMNN | 21.75 | 21.736 | 0.1 | 0.1 | 55.1921 |
| | GNB | 19.35 | **19.075** | - | - | 0.5492 |
| | SVM | **19.34** | 19.082 | - | - | 1.6912 |
| | DT | 26.94 | 27.014 | - | - | 0.5703 |
| 16 | He-MRHGRC | 30.11 | 30.996 | 0.1 | 0.4 | 1.2686 |
| | Ho-MRHGRC | 28.70 | 30.564 | 0.1 | 0.1 | 1.8852 |
| | GFMM | 32.66 | 33.415 | 0.1 | 0.1 | 18.0554 |
| | FMNN | 29.78 | 31.035 | 0.1 | 0.1 | 69.6761 |
| | KNEFMNN | 33.42 | 34.670 | 0.1 | 0.1 | 22.3418 |
| | EFMNN | 31.80 | 33.239 | 0.1 | 0.1 | 76.0920 |
| | GNB | **27.12** | 28.190 | - | - | 0.5764 |
| | SVM | 27.29 | **28.103** | - | - | 1.6455 |
| | DT | 38.813 | 39.644 | - | - | 0.6023 |

of our proposed approach.

***Increase the number of classes:***

The purpose of the experiment in this subsection is to evaluate the performance of the proposed method on multi-class datasets. We kept the number of dimensions $n = 2$, the number of samples $N = 10,000$, and only changed the number of classes to form synthetic multi-class datasets with $C \in \{2, 4, 16\}$. The covariance matrices stay the same as in the case of changing the number of samples.

Fig. 4 shows the change in error rates of fuzzy min-max classifiers with a different number of classes on the testing sets. It can be easily seen that the error rates of our method are lowest compared to other fuzzy min-max neural networks on all multi-class synthetic datasets at high abstraction levels of granular representations. At high abstraction levels, the error rates of other fuzzy min-max neural networks increase rapidly, while the error rate of our classifier still maintains the stability. In addition, the error rates of our method also slowly augment in contrast to the behaviors of other considered types of fuzzy min-max neural networks when increasing the abstraction level of granular representations. These facts demonstrate the efficiency of our proposed method on multi-class datasets. The lowest error rates of classifiers on validation and testing sets, as well as total training time, are shown in Table II. It is observed that the predictive accuracy of our method outperforms all considered types of fuzzy min-max classifiers and decision tree, but it cannot overcome the Gaussian Naive Bayes and support vector machine methods. The training time of our method is faster than other fuzzy min-max neural networks and support vector machines on the considered multi-class synthetic datasets.

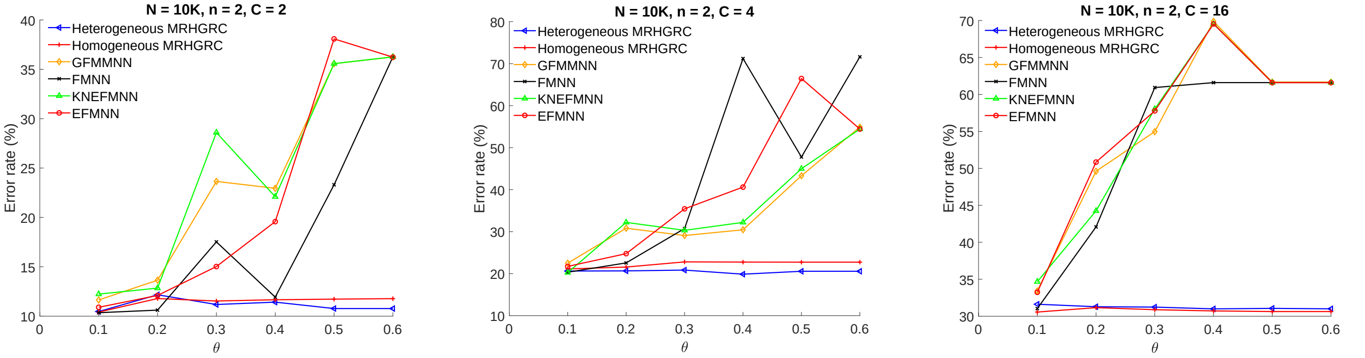
***Increase the number of features:***

To generate the multi-dimensional synthetic datasets with the number of samples $N = 10K$ and the number of classes $C = 2$, we used the similar settings as in generation of datasets with different number of samples. The means of classes are $\mu_1 = [0, \ldots, 0]^T, \mu_2 = [2.56, 0, \ldots, 0]^T$, and the covariance matrices are as follows:

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 \end{bmatrix}$$

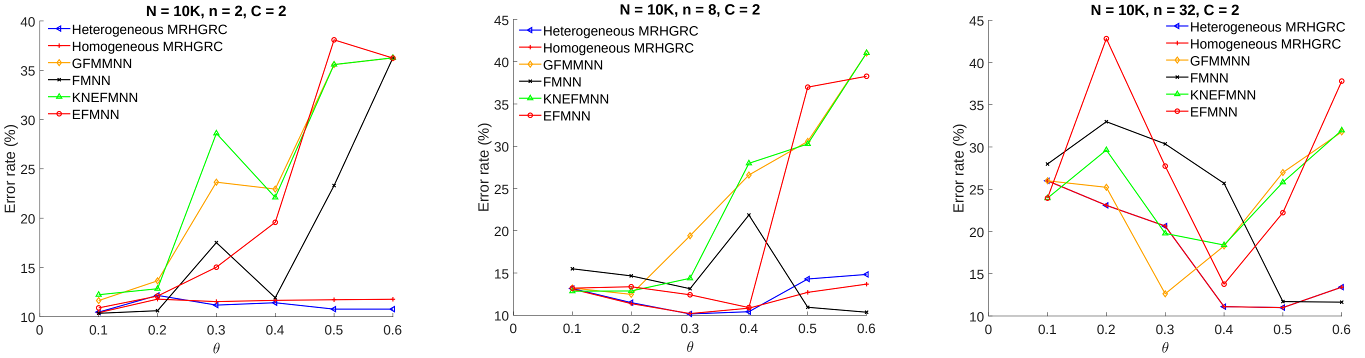
The size of each expression corresponds to the number of dimensions $n$ of the problem. Fukunaga [35] stated that the general Bayes error of 10% and this Bayes error stays the same even when $n$ changes.

Fig. 5 shows the change in the error rates with different levels of granularity on multi-dimensional synthetic datasets. In general, with a low number of dimensions, our method outperforms other fuzzy min-max neural networks. With high dimensionality and a small number of samples, the high levels of granularity result in high error rates, and misclassification results considerably drops when the value of $\theta$ increases. The same trend also happens to the FMNN when its accuracy at $\theta = 0.5$ or $\theta = 0.6$ is quite high. Apart from the FMNN on high dimensional datasets, our proposed method is better than three other fuzzy min-max classifiers at high abstraction levels. Table III reports the lowest error rates of classifiers on validation and testing multi-dimensional sets as well as the



(a) 2 classes (b) 4 classes (c) 16 classes

Fig. 4. The error rate of classifiers on synthetic linear boundary datasets with the different number of classes.



(a) 2 features (b) 8 features (c) 32 features

Fig. 5. The error rate of classifiers on synthetic linear boundary datasets with the different number of features.

TABLE III
THE LOWEST ERROR RATES AND TRAINING TIME OF CLASSIFIERS ON SYNTHETIC LINEAR BOUNDARY DATASETS WITH DIFFERENT FEATURES ($N = 10K, C = 2$)

| n | Algorithm | min $E_V$ | min $E_T$ | $\theta_V$ | $\theta_T$ | Time (s) |
|---|---|---|---|---|---|---|
| 2 | He-MRHGRC | 10.250 | 10.467 | 0.1 | 0.1 | 1.1378 |
| | Ho-MRHGRC | 10.100 | 10.413 | 0.1 | 0.1 | 1.3215 |
| | GFMM | 11.540 | 11.639 | 0.1 | 0.1 | 8.6718 |
| | FMNN | 10.050 | 10.349 | 0.1 | 0.1 | 46.4789 |
| | KNEFMNN | 12.070 | 12.232 | 0.1 | 0.1 | 9.4459 |
| | EFMNN | 10.440 | 10.897 | 0.1 | 0.1 | 48.9892 |
| | GNB | **9.850** | **9.964** | - | - | 0.5218 |
| | SVM | 9.910 | 9.983 | - | - | 1.5468 |
| | DT | 15.330 | 14.861 | - | - | 0.5405 |
| 8 | He-MRHGRC | 10.330 | 10.153 | 0.3 | 0.3 | 21.9131 |
| | Ho-MRHGRC | 10.460 | 10.201 | 0.3 | 0.3 | 23.0554 |
| | GFMM | 12.170 | 12.474 | 0.1 | 0.2 | 196.0682 |
| | FMNN | 10.250 | 10.360 | 0.6 | 0.6 | 302.8683 |
| | KNEFMNN | 12.720 | 12.844 | 0.1 | 0.1 | 618.2524 |
| | EFMNN | 11.300 | 10.907 | 0.4 | 0.4 | 579.3113 |
| | GNB | **9.940** | **9.919** | - | - | 0.5915 |
| | SVM | 9.980 | 9.927 | - | - | 2.0801 |
| | DT | 15.383 | 15.087 | - | - | 0.6769 |
| 32 | He-MRHGRC | 11.070 | 10.995 | 0.5 | 0.5 | 226.3193 |
| | Ho-MRHGRC | 11.070 | 10.995 | 0.5 | 0.5 | 226.0611 |
| | GFMM | 12.390 | 12.625 | 0.3 | 0.3 | 847.6977 |
| | FMNN | 11.830 | 11.637 | 0.5 | 0.6 | 1113.6836 |
| | KNEFMNN | 17.410 | 18.395 | 0.1 | 0.4 | 837.9571 |
| | EFMNN | 13.890 | 13.766 | 0.4 | 0.4 | 1114.4976 |
| | GNB | 10.280 | 10.088 | - | - | 0.7154 |
| | SVM | **10.220** | **10.079** | - | - | 4.5937 |
| | DT | 15.400 | 15.201 | - | - | 1.0960 |

total training time through six abstraction levels of granular representations. The training time of our method is much faster than other types of fuzzy min-max neural networks. Generally, the performance of our proposed method overcomes the decision tree and other types of fuzzy min-max neural networks, but its predictive results cannot defeat the Gaussian Naive Bayes and support vector machines. It can be observed that the best performance on validation and testing sets obtains at the same abstraction level of granular representations on all considered multi-dimensional datasets. This fact indicates that we can use the validation set to choose the best classifier at a given abstraction level among constructed models through different granularity levels.

*2) Non-linear Boundary:* To generate non-linear boundary datasets, we set up the Gaussian means of the first class: $\mu_1 = [-2, 1.5]^T, \mu_2 = [1.5, 1]^T$ and the Gaussian means of the second class: $\mu_3 = [-1.5, 3]^T, \mu_4 = [1.5, 2.5]^T$. The covariance matrices for the first class $\Sigma_1, \Sigma_2$ and for the second class $\Sigma_3, \Sigma_4$ were established as follows:

$$\Sigma_1 = \begin{bmatrix} 0.5 & 0.05 \\ 0.05 & 0.4 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.5 & 0.05 \\ 0.05 & 0.3 \end{bmatrix},$$
$$\Sigma_3 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, \Sigma_4 = \begin{bmatrix} 0.5 & 0.05 \\ 0.05 & 0.2 \end{bmatrix},$$

The number of samples for each class was equal, and the generated samples were normalized to the range of [0, 1]. We created only a testing set including 100,000 samples and a validation set with 10,000 patterns. Three different training sets containing 10K, 100K, and 5M samples were used to train classifiers. We aim to evaluate the predictive results of our method on the non-linear boundary dataset when changing the sizes of the training set.

Fig. 6 shows the changes in the error rates through different levels of granularity of classifiers on non-linear boundary datasets. It can be observed that the error rates of our proposed method trained on the large-sized non-linear boundary datasets are better than those using other types of fuzzy min-max neural networks, especially at high abstraction levels of granular representations. While other fuzzy min-max neural networks show the increase in the error rates if the value of $\theta$ grows up, our method is capable of maintaining the stability of predictive results even in the case of high abstraction levels. When the number of samples increases, the error rates of other fuzzy min-max classifiers usually rise, whereas the error rate in our approach only fluctuates a little. These results indicate that our approach may reduce the influence of overfitting because of constructing higher abstraction level of granular data representations using the learned knowledge from lower abstraction levels.

The best performance of our approach does not often happen at the smallest value of $\theta$ on these non-linear datasets. Results regarding accuracy on validation and testing sets reported in Table IV confirm this statement. These figures also illustrate the effectiveness of the processing steps in phase 2. Unlike the linear boundary datasets, our method overcomes the Gaussian Naive Bayes to become two best classifiers (along with SVM) among classifiers considered. Although SVM outperformed our approach, its runtime on large-sized datasets is much slower than our method. The training time of our algorithm is much faster than other types of fuzzy min-max neural networks and SVM, but it is still slower than Gaussian Naive Bayes and decision tree techniques.

### B. *Performance of the Proposed Method on Real Datasets*

Aiming to attain the fairest comparison, we used 12 datasets with diverse ranges of the number of sizes, dimensions, and classes. These datasets were taken from the LIBSVM [36], Kaggle [37], and UCI repositories [38] and their properties are described in Table V. For the *SUSY* dataset, the last 500,000 patterns were used for the test set as shown in [39]. From the results of synthetic datasets, we can see that the performance of the multi-resolution hierarchical granular representation based classifier using the heterogeneous data distribution technique is more stable than that utilizing the homogeneous distribution method. Therefore, the experiments in the rest of this paper were conducted for only the heterogeneous classifier.

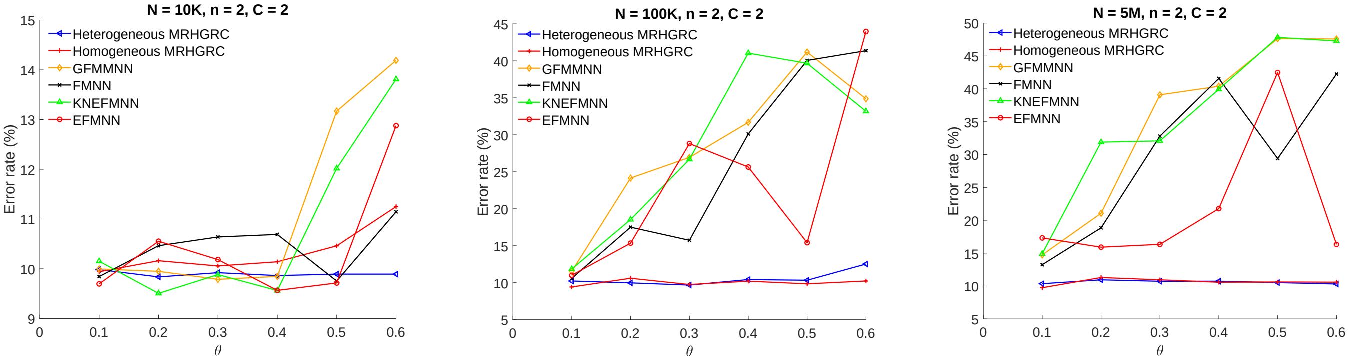
Table VI shows the number of generated hyperboxes for the He-MRHGRC on real datasets at different abstraction levels of granular representations. It can be seen that the number of hyperboxes at the value of $\theta = 0.6$ is significantly reduced in comparison to those at $\theta = 0.1$. However, the error rates of the classifiers on testing sets at $\theta = 0.6$ do not change so much compared to those at $\theta = 0.1$. This fact is illustrated in Fig. 7 and figures in the supplemental file. From these figures, it is observed that at the high values of maximum hyperbox size such as $\theta = 0.5$ and $\theta = 0.6$, our classifier achieves the best performance compared to other considered types of fuzzy min-max neural networks. We can also observe that the prediction accuracy of our method is usually much better than that using other types of fuzzy min-max classifiers on most of the data granulation levels. The error rate of our classifier regularly increases slowly with the increase in the abstraction level of granules, even in some cases, the error rate declines at a high abstraction level of granular representations. The best performance of classifiers on validation and testing sets, as well as training time through six granularity levels, are reported in the supplemental file.

TABLE IV
THE LOWEST ERROR RATES AND TRAINING TIME OF CLASSIFIERS ON SYNTHETIC NON-LINEAR BOUNDARY DATASETS WITH DIFFERENT NUMBER OF SAMPLES ($n = 2, C = 2$)

| **N** | **Algorithm** | $\min E_V$ | $\min E_T$ | $\theta_V$ | $\theta_T$ | **Time (s)** |
|---|---|---|---|---|---|---|
| 10K | He-MRHGRC | 9.950 | 9.836 | 0.2 | 0.2 | 0.9616 |
| | Ho-MRHGRC | 9.820 | 9.940 | 0.1 | 0.1 | 1.1070 |
| | GFMM | 10.200 | 9.787 | 0.4 | 0.5 | 10.5495 |
| | FMNN | 9.770 | 9.753 | 0.5 | 0.5 | 61.1130 |
| | KNEFMNN | 9.890 | 9.505 | 0.2 | 0.2 | 16.1099 |
| | EFMNN | **9.750** | 9.565 | 0.1 | 0.4 | 60.6073 |
| | GNB | 10.740 | 10.626 | - | - | 0.5218 |
| | SVM | **9.750** | **9.490** | - | - | 1.5565 |
| | DT | 14.107 | 13.831 | - | - | 0.5388 |
| 100K | He-MRHGRC | 10.130 | 9.670 | 0.3 | 0.3 | 2.5310 |
| | Ho-MRHGRC | 9.910 | 9.412 | 0.1 | 0.1 | 2.3560 |
| | GFMM | 11.810 | 11.520 | 0.1 | 0.1 | 44.7778 |
| | FMNN | 10.880 | 10.575 | 0.1 | 0.1 | 588.4412 |
| | KNEFMNN | 12.470 | 11.836 | 0.1 | 0.1 | 42.9151 |
| | EFMNN | 11.020 | 10.992 | 0.1 | 0.1 | 485.7613 |
| | GNB | 10.830 | 10.702 | - | - | 0.9006 |
| | SVM | **9.650** | **9.338** | - | - | 93.4474 |
| | DT | 14.277 | 13.642 | - | - | 1.1767 |
| 5M | He-MRHGRC | 10.370 | 10.306 | 0.1 | 0.6 | 91.7894 |
| | Ho-MRHGRC | **9.940** | **9.737** | 0.1 | 0.1 | 69.5106 |
| | GFMM | 15.260 | 14.730 | 0.1 | 0.1 | 1927.6191 |
| | FMNN | 13.160 | 13.243 | 0.1 | 0.1 | 53274.4387 |
| | KNEFMNN | 15.040 | 14.905 | 0.1 | 0.1 | 1551.5220 |
| | EFMNN | 15.660 | 15.907 | 0.2 | 0.2 | 54487.6978 |
| | GNB | 10.840 | 10.690 | - | - | 22.9849 |
| | SVM | N/A | N/A | - | - | N/A |
| | DT | 13.790 | 13.645 | - | - | 49.9919 |

Although our method cannot achieve the best classification accuracy on all considered datasets, its performance is located in the top 2 for all datasets. The Gaussian Naive Bayes classifiers obtained the best predictive results on synthetic linear boundary datasets, but it fell to the last position and became the worst classifier on real datasets because real datasets are highly non-linear. On datasets with highly non-linear decision boundaries such as *covtype*, *PhysioNet MIT-BIH Arrhythmia*, and *MiniBooNE*, our proposed method still produces the good predictive accuracy.

The training process of our method is much faster than other types of fuzzy min-max neural networks on all considered datasets. Notably, on some large-sized complex datasets such as *covtype* and *SUSY*, the training time of other fuzzy min-max classifiers is costly, but their accuracy is worse than our method, which takes less training time. Our approach is frequently faster than SVM and can deal with datasets with millions of samples, while the SVM approach cannot perform.



(a) 10K samples

(b) 100K samples

(c) 5M samples

Fig. 6. The error rate of classifiers on synthetic non-linear boundary datasets with the different number of samples.

TABLE V
The Real Datasets and Their Statistics

| Dataset | #Dimensions | #Classes | #Training samples | #Validation samples | #Testing samples | Source |
|---|---|---|---|---|---|---|
| Poker Hand | 10 | 10 | 25,010 | 50,000 | 950000 | LIBSVM |
| SensIT Vehicle | 100 | 3 | 68,970 | 9,852 | 19,706 | LIBSVM |
| Skin_NonSkin | 3 | 2 | 171,540 | 24,260 | 49,257 | LIBSVM |
| Covtype | 54 | 7 | 406,709 | 58,095 | 116,208 | LIBSVM |
| White wine quality | 11 | 7 | 2,449 | 1,224 | 1,225 | Kaggle |
| PhysioNet MIT-BIH Arrhythmia | 187 | 5 | 74,421 | 13,133 | 21,892 | Kaggle |
| MAGIC Gamma Telescope | 10 | 2 | 11,887 | 3,567 | 3,566 | UCI |
| Letter | 16 | 26 | 15,312 | 2,188 | 2,500 | UCI |
| Default of credit card clients | 23 | 2 | 18,750 | 5,625 | 5,625 | UCI |
| MoCap Hand Postures | 36 | 5 | 53,104 | 9,371 | 15,620 | UCI |
| MiniBooNE | 50 | 2 | 91,044 | 12,877 | 26,143 | UCI |
| SUSY | 18 | 2 | 4,400,000 | 100,000 | 500,000 | UCI |

TABLE VI
The Change in the Number of Generated Hyperboxes Through Different Levels of Granularity of the Proposed Method

| Dataset | θ | | | | | |
|---|---|---|---|---|---|---|
| | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| Skin_NonSkin | 1012 | 248 | 127 | 85 | 64 | 51 |
| Poker Hand | 11563 | 11414 | 10905 | 3776 | 2939 | 2610 |
| Covtype | 94026 | 13560 | 5224 | 2391 | 1330 | 846 |
| SensIT Vehicle | 5526 | 2139 | 1048 | 667 | 523 | 457 |
| PhysioNet MIT-BIH Arrhythmia | 60990 | 26420 | 15352 | 8689 | 5261 | 3241 |
| White wine quality | 1531 | 676 | 599 | 559 | 544 | 526 |
| Default of credit card clients | 2421 | 529 | 337 | 76 | 48 | 29 |
| Letter | 9236 | 1677 | 952 | 646 | 595 | 556 |
| MAGIC Gamma Telescope | 1439 | 691 | 471 | 384 | 335 | 308 |
| MiniBooNE | 444 | 104 | 24 | 10 | 6 | 6 |
| SUSY | - | - | 26187 | 25867 | 16754 | 13017 |

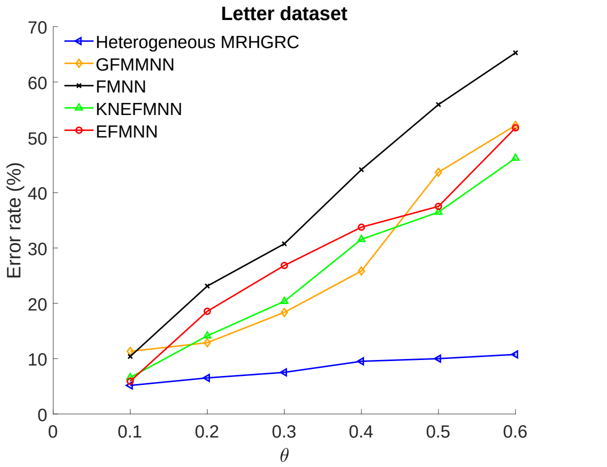


Fig. 7. The error rate of classifiers on the *Letter* datasets through data abstraction levels.

On many datasets, the best predictive results on validation and testing sets were achieved at the same abstraction level of granular representations. In the case that the best model on the validation set has different abstraction level compared to the best model on the testing set, the error rate on the testing set if using the best classifier on the validation set is also near the minimum error. These figures show that our proposed method is stable, and it can achieve a high predictive accuracy on both synthetic and real datasets.

### *C. The Vital Role of the Pruning Process and the Use of Sample Centroids*

This experiment aims to assess the important roles of the pruning process and the use of sample centroids on the performance of the proposed method. The experimental results related to these issues are presented in Table VII. It is easily observed that the pruning step contributes to significantly reducing the number of generated hyperboxes, especially in

TABLE VII
THE ROLE OF THE PRUNING PROCESS AND THE USE OF SAMPLE CENTROIDS

| **Dataset** | **Num hyperboxes** | | **Error rate before pruning (%)** | **Error rate after pruning (%)** | **No. of predicted samples using centroids before pruning** | | **No. of predicted samples using centroids after pruning** | |
|---|---|---|---|---|---|---|---|---|
| | **Before pruning** | **After pruning** | | | **Total** | **Wrong** | **Total** | **Wrong** |
| Skin_NonSkin | 1,358 | 1,012 | 0.1726 | 0.0974 | 1,509 | 73 | 594 | 30 |
| Poker Hand | 24,991 | 11,563 | 53.5951 | 49.8128 | 600,804 | 322,962 | 725,314 | 362,196 |
| SensIT Vehicle | 61,391 | 5,526 | 23.6730 | 20.9073 | 2 | 1 | 0 | 0 |
| Default of credit card clients | 9,256 | 2,421 | 22.3822 | 19.7689 | 662 | 291 | 312 | 127 |
| Covtype | 95971 | 94026 | 7.7335 | 7.5356 | 2700 | 975 | 2213 | 783 |
| PhysioNet MIT-BIH Arrhythmia | 61,419 | 60,990 | 3.6589 | 3.5492 | 49 | 9 | 48 | 8 |
| MiniBooNE | 1,079 | 444 | 16.4289 | 13.9043 | 14,947 | 3,404 | 11,205 | 2,575 |
| SUSY | 55,096 | 26,187 | 30.8548 | 28.3456 | 410,094 | 145,709 | 370,570 | 124,850 |

*SensIT Vehicle*, *Default of credit card clients*, *SUSY* datasets. When the poorly performing hyperboxes are removed, the accuracy of the model increases considerably. These figures indicate the critical role of the pruning process with regards to reducing the complexity of the model and enhancing the predictive performance.

We can also see that the use of sample centroids and Euclidean distance may predict accurately from 50% to 95% of the samples located in the overlapping regions between different classes. The predictive accuracy depends on the distribution and complexity of underlying data. With the use of sample centroids, we do not need to use the overlap test and contraction process in phase 1 at the highest level of granularity. This strategy leads to accelerating the training process of the proposed method compared to other types of fuzzy min-max neural networks, especially in large-sized datasets such as *covtype* or *SUSY*. These facts point to the effectiveness of the pruning process and the usage of sample centroids on improving the performance of our approach in terms of both accuracy and training time.

### *D. Ability to Handling Missing Values*

This experiment was conducted on two datasets containing many missing values, i.e., *PhysioNet MIT-BIH Arrhythmia* and *MoCap Hand Postures* datasets. The aim of this experiment is to demonstrate the ability to handle missing values of our method to preserve the uncertainty of input data without doing any pre-processing steps. We also generated three other training datasets from the original data by replacing missing values with the zero, mean, or median value of each feature. Then, these values were used to fill in the missing values of corresponding features in the testing and validation sets. The obtained results are presented in Table VIII. The predictive accuracy of the classifier trained on the datasets with missing values cannot be superior to ones trained on the datasets imputed by the median, mean or zero values. However, the training time is reduced, and the characteristic of the proposed method is still preserved, in which the accuracy of the classifier is maintained at high levels of abstraction, and its behavior is nearly the same on both validation and testing sets. The replacement of missing values by other values is usually biased and inflexible in real-world applications. The capability of deducing directly from data with missing values ensures the maintenance of the online learning property of the fuzzy min-max neural network on the incomplete input data.

TABLE VIII
THE TRAINING TIME AND THE LOWEST ERROR RATES OF OUR METHOD ON THE DATASETS WITH MISSING VALUES

| **Dataset** | **Training time (s)** | $\min E_V$ | $\min E_T$ |
|---|---|---|---|
| Arrhythmia with replacing missing values by zero values | 53,100.2895 | 3.0762<br>$(\theta = 0.1)$ | 3.5492<br>$(\theta = 0.1)$ |
| Arrhythmia with replacing missing values by mean values | 60,980.5110 | 2.6879<br>$(\theta = 0.1)$ | 3.3848<br>$(\theta = 0.1)$ |
| Arrhythmia with replacing missing values by median values | 60,570.4315 | 2.7031<br>$(\theta = 0.1)$ | 3.2980<br>$(\theta = 0.2)$ |
| Arrhythmia with missing values retained | 58,188.8138 | 2.6955<br>$(\theta = 0.1)$ | 3.1473<br>$(\theta = 0.1)$ |
| Postures with replacing missing values by zero values | 5,845.9722 | 6.6482<br>$(\theta = 0.1)$ | 7.7529<br>$(\theta = 0.4)$ |
| Postures with replacing missing values by mean values | 5,343.0038 | 8.5370<br>$(\theta = 0.1)$ | 9.7631<br>$(\theta = 0.3)$ |
| Postures with replacing missing values by median values | 4,914.4475 | 8.4089<br>$(\theta = 0.1)$ | 9.9936<br>$(\theta = 0.3)$ |
| Postures with missing values retained | 2,153.8121 | 14.5662<br>$(\theta = 0.4)$ | 13.7900<br>$(\theta = 0.4)$ |

### *E. Comparison to State-of-the-art Studies*

The purpose of this section is to compare our method with recent studies of classification algorithms on large-sized datasets in physics and medical diagnostics. The first experiment was performed on the *SUSY* dataset to distinguish between a signal process producing super-symmetric particles and a background process. To attain this purpose, Baldi et al. [39] compared the performance of a deep neural network with boosted decision trees using the area under the curve (AUC) metrics. In another study, Sakai et al. [40] evaluated different methods of AUC optimization in combination with support vector machines to enhance the efficiency of the final predictive model. The AUC values of these studies along with our method are reported in Table IX. It can be seen that our approach overcomes all approaches in Sakai's research, but it cannot outperform the deep learning methods and boosted trees on the considered dataset.

The second experiment was conducted on a medical dataset (*PhysioNet MIT-BIH Arrhythmia*) containing Electrocardiogram (ECG) signal used for the classification of heartbeats. There are many studies on ECG heartbeat classification such as deep residual convolution neural network [41], a 9-layer deep convolutional neural network on the augmentation of the original data [42], combinations of a discrete wavelet transform with neural networks, SVM [43], and random forest [44]. The *PhysioNet MIT-BIH Arrhythmia* dataset contains

TABLE IX
THE AUC VALUE OF THE PROPOSED METHOD AND OTHER METHODS ON THE *SUSY* DATASET

| **Method** | **AUC** |
|---|---|
| Boosted decision tree [39] | 0.863 |
| Deep neural network [39] | 0.876 |
| Deep neural network with dropout [39] | **0.879** |
| Positive-Negative and unlabeled data based AUC optimization [40] | 0.647 |
| Semi-supervised rankboost based AUC optimization [40] | 0.709 |
| Semi-supervised AUC-optimized logistic sigmoid [40] | 0.556 |
| Optimum AUC with a generative model [40] | 0.577 |
| He-MRHGRC (Our method) | 0.799 |

TABLE X
THE ACCURACY OF THE PROPOSED METHOD AND OTHER METHODS ON THE *PhysioNet MIT-BIH Arrhythmia* DATASET

| **Method** | **Accuracy(%)** |
|---|---|
| Deep residual Convolutional neural network [41] | 93.4 |
| Augmentation + Deep convolutional neural network [42] | 93.5 |
| Discrete wavelet transform + SVM [43] | 93.8 |
| Discrete wavelet transform + NN [43] | 94.52 |
| Discrete wavelet transform + Random Forest [44] | 94.6 |
| Our method on the dataset with the missing values | **96.85** |
| Our method on the dataset with zero padding | 96.45 |

many missing values and above studies used the zero padding mechanism for these values. Our method can directly handle missing values without any imputations. The accuracy of our method on the datasets with missing values and zero paddings is shown in Table X along with results taken from other studies. It is observed that our approach on the dataset including missing values outperforms all other methods considered. From these comparisons, we can conclude that our proposed method is extremely competitive to other state-of-the-art studies published on real datasets.

## V. CONCLUSION AND FUTURE WORK

This paper presented a method to construct classification models based on multi-resolution hierarchical granular representations using hyperbox fuzzy sets. Our approach can maintain good classification accuracy at high abstraction levels with a low number of hyperboxes. The best classifier on the validation set usually produces the best predictive results on unseen data as well. One of the interesting characteristics of our method is the capability of handling missing values without the need for missing values imputation. This property makes it flexible for real-world applications, where the data incompleteness usually occurs. In general, our method outperformed other typical types of fuzzy min-max neural networks using the contraction process for dealing with overlapping regions in terms of both accuracy and training time. Furthermore, our proposed technique can be scaled to large-sized datasets based on the parallel execution of the hyperbox building process at the highest level of granularity to form core hyperboxes from sample points rapidly. These hyperboxes are then refined at higher abstraction levels to reduce the complexity and maintain consistent predictive performance.

The patterns located in the overlapping regions are currently classified by using Euclidean distance to the sample centroids. Future work will focus on deploying the probability estimation measure to deal with these samples. The predictive results of the proposed method depend on the order of presentations of the training patterns because it is based on the online learning ability of the general fuzzy min-max neural network. In addition, the proposed method is sensitive to noise and outliers as well. In real-world applications, noisy data are frequently encountered; thus they can lead to serious stability issue. Therefore, outlier detection and noise removal are essential issues which need to be tackled in future work. Furthermore, we also intend to combine hyperboxes generated in different levels of granularity to build an optimal ensemble model for pattern recognition.

## ACKNOWLEDGMENT

T.T. Khuat acknowledges FEIT-UTS for awarding his PhD scholarships (IRS and FEIT scholarships).

## REFERENCES

[1] G. Wang, J. Yang, and J. Xu, “Granular computing: from granularity optimization to multi-granularity joint problem solving,” *Granular Computing*, vol. 2, no. 3, pp. 105–120, 2017.

[2] J. A. Morente-Molinera, J. Mezei, C. Carlsson, and E. Herrera-Viedma, “Improving supervised learning classification methods using multigranular linguistic modeling and fuzzy entropy,” *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1078–1089, 2017.

[3] L. A. Zadeh, “Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic,” *Fuzzy Sets and Systems*, vol. 90, no. 2, pp. 111 – 127, 1997.

[4] W. Pedrycz, G. Succi, A. Sillitti, and J. Iljazi, “Data description: A general framework of information granules,” *Knowledge-Based Systems*, vol. 80, pp. 98 – 108, 2015.

[5] W. Pedrycz, “Allocation of information granularity in optimization and decision-making models: Towards building the foundations of granular computing,” *European Journal of Operational Research*, vol. 232, no. 1, pp. 137 – 145, 2014.

[6] J. Xu, G. Wang, T. Li, and W. Pedrycz, “Local-density-based optimal granulation and manifold information granule description,” *IEEE Transactions on Cybernetics*, vol. 48, no. 10, pp. 2795–2808, 2018.

[7] W. Xu and J. Yu, “A novel approach to information fusion in multi-source datasets: A granular computing viewpoint,” *Information Sciences*, vol. 378, pp. 410 – 423, 2017.

[8] R. Al-Hmouz, W. Pedrycz, A. S. Balamash, and A. Morfeq, “Granular description of data in a non-stationary environment,” *Soft Computing*, vol. 22, no. 2, pp. 523–540, 2018.

[9] C. P. Chen and C.-Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on big data,” *Information Sciences*, vol. 275, pp. 314 – 347, 2014.

[10] T. T. Khuat, D. Ruta, and B. Gabrys, “Hyperbox based machine learning algorithms: A comprehensive survey,” *arXiv e-prints*, p. arXiv:1901.11303, 2019.

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

[12] 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.

[13] J. de Jesús Rubio, “Sofmls: online self-organizing fuzzy modified least-squares network,” *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1296–1309, 2009.

[14] X.-M. Zhang and Q.-L. Han, “State estimation for static neural networks with time-varying delays based on an improved reciprocally convex inequality,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 1376–1381, 2017.

[15] J. de Jesús Rubio, “A method with neural networks for the classification of fruits and vegetables,” *Soft Computing*, vol. 21, no. 23, pp. 7207–7220, 2017.

[16] M.-Y. Cheng, D. Prayogo, and Y.-W. Wu, “Prediction of permanent deformation in asphalt pavements using a novel symbiotic organisms search–least squares support vector regression,” *Neural Computing and Applications*, pp. 1–13, 2018.

[17] J. d. J. Rubio, D. Ricardo Cruz, I. Elias, G. Ochoa, R. Balcazarand, and A. Aguilar, “Anfis system for classification of brain signals,” *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–9, 2019.

[18] B. Gabrys, “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.

[19] W. Pedrycz, “Granular computing for data analytics: A manifesto of human-centric computing,” *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 6, pp. 1025–1034, 2018.

[20] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2009.

[21] L. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[22] W. Pedrycz, “Interpretation of clusters in the framework of shadowed sets,” *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2439 – 2449, 2005.

[23] Z. Pawlak and A. Skowron, “Rough sets and boolean reasoning,” *Information Sciences*, vol. 177, no. 1, pp. 41–73, 2007.

[24] W. Pedrycz and W. Homenda, “Building the fundamentals of granular computing: A principle of justifiable granularity,” *Applied Soft Computing*, vol. 13, no. 10, pp. 4209 – 4218, 2013.

[25] W. Pedrycz and A. Bargiela, “An optimization of allocation of information granularity in the interpretation of data structures: Toward granular fuzzy clustering,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 3, pp. 582–590, 2012.

[26] Z. Sahel, A. Bouchachia, B. Gabrys, and P. Rogers, “Adaptive mechanisms for classification problems with drifting data,” in *Knowledge-Based Intelligent Information and Engineering Systems*, B. Apolloni, R. J. Howlett, and L. Jain, Eds. Springer Berlin Heidelberg, 2007, pp. 419–426.

[27] G. Peters and R. Weber, “Dcc: a framework for dynamic granular clustering,” *Granular Computing*, vol. 1, no. 1, pp. 1–11, 2016.

[28] 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.

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

[30] ——, “Neuro-fuzzy approach to processing inputs with missing values in pattern recognition problems,” *International Journal of Approximate Reasoning*, vol. 30, no. 3, pp. 149–179, 2002.

[31] N. Macia, E. Bernado-Mansilla, A. Orriols-Puig, and T. K. Ho, “Learner excellence biased by data set selection: A case for data characterisation and artificial data sets,” *Pattern Recognition*, vol. 46, no. 3, pp. 1054 – 1066, 2013.

[32] M. Budka and B. Gabrys, “Density-preserving sampling: Robust and efficient alternative to cross-validation for error estimation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 1, pp. 22–34, 2013.

[33] M. F. Mohammed and C. P. Lim, “An enhanced fuzzy min–max neural network for pattern classification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 417–429, 2015.

[34] M. F. Mohammed and C. P. Lim, “Improving the fuzzy min-max neural network with a k-nearest hyperbox expansion rule for pattern classification,” *Appl. Soft Comput.*, vol. 52, no. C, pp. 135–145, 2017.

[35] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[36] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[37] Kaggle, “Kaggle datasets,” 2019. [Online]. Available: https://www.kaggle.com/datasets

[38] D. Dua and E. Karra Taniskidou, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[39] P. Baldi, P. Sadowski, and D. Whiteson, “Searching for exotic particles in high-energy physics with deep learning,” *Nature Communications*, vol. 5, p. 4308, 2014.

[40] T. Sakai, G. Niu, and M. Sugiyama, “Semi-supervised auc optimization based on positive-unlabeled learning,” *Machine Learning*, vol. 107, no. 4, pp. 767–794, 2018.

[41] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, “Ecg heartbeat classification: A deep transferable representation,” in *IEEE International Conference on Healthcare Informatics (ICHI)*, 2018, pp. 443–444.

[42] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and R. S. Tan, “A deep convolutional neural network model to classify heartbeats,” *Computers in Biology and Medicine*, vol. 89, pp. 389 – 396, 2017.

[43] R. J. Martis, U. R. Acharya, C. M. Lim, K. M. Mandana, A. K. Ray, and C. Chakraborty, “Application of higher order cumulant features for cardiac health diagnosis using ecg signals,” *International Journal of Neural Systems*, vol. 23, no. 04, p. 1350014, 2013.

[44] T. Li and M. Zhou, “Ecg classification using wavelet packet entropy and random forests,” *Entropy*, vol. 18, no. 8, 2016.

**Thanh Tung Khuat** received the B.E degree in Software Engineering from University of Science and Technology, Danang, Vietnam, in 2014. Currently, he is working towards the Ph.D. degree at the Advanced Analytics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia. His research interests include machine learning, fuzzy systems, knowledge discovery, evolutionary computation, intelligent optimization techniques and applications in software engineering. He has authored and co-authored over 20 peer-reviewed publications in the areas of machine learning and computational intelligence.

**Fang Chen** is the Executive Director Data Science and a Distinguished Professor with the University of Technology Sydney, Ultimo, NSW, Australia. She is a Thought Leader in AI and data science. She has created many world-class AI innovations while working with Beijing Jiaotong University, Intel, Motorola, NICTA, and CSIRO, and helped governments and industries utilising data and significantly increasing productivity, safety, and customer satisfaction. Through impactful successes, she gained many recognitions, such as the ITS Australia National Award 2014 and 2015, and NSW iAwards 2017. She is the NSW Water Professional of the Year 2016, the National and NSW Research, and the Innovation Award by Australian Water association. She was the recipient of the “Brian Shackle Award” 2017 for the most outstanding contribution with international impact in the field of human interaction with computers and information technology. She is the recipient of the Oscar Prize in Australian science-Australian Museum Eureka Prize 2018 for Excellence in Data Science. She has 280 publications and 30 patents in 8 countries.

**Bogdan Gabrys** received the M.Sc. degree in electronics and telecommunication from Silesian Technical University, Gliwice, Poland, in 1994, and the Ph.D. degree in computer science from Nottingham Trent University, Nottingham, U.K., in 1998.

Over the last 25 years, he has been working at various universities and research and development departments of commercial institutions. He is currently a Professor of Data Science and a Director of the Advanced Analytics Institute at the University of Technology Sydney, Sydney, Australia. His research activities have concentrated on the areas of data science, complex adaptive systems, computational intelligence, machine learning, predictive analytics, and their diverse applications. He has published over 180 research papers, chaired conferences, workshops, and special sessions, and been on program committees of a large number of international conferences with the data science, computational intelligence, machine learning, and data mining themes. He is also a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), a Memebr of IEEE Computational Intelligence Society and a Fellow of the Higher Education Academy (HEA) in the UK. He is frequently invited to give keynote and plenary talks at international conferences and lectures at internationally leading research centres and commercial research labs. More details can be found at: http://bogdan-gabrys.com