Multiclass Classification With Fuzzy-Feature Observations: Theory and Algorithms

Guangzhi Ma, Jie Lu, *Fellow, IEEE*, Feng Liu, *Member, IEEE*,
Zhen Fang, *Member, IEEE*, and Guangquan Zhang

***Abstract*—The theoretical analysis of multiclass classification has proved that the existing multiclass classification methods can train a classifier with high classification accuracy on the test set, when the instances are *precise* in the training and test sets with same distribution and enough instances can be collected in the training set. However, one limitation with multiclass classification has not been solved: how to improve the classification accuracy of multiclass classification problems when only imprecise observations are available. Hence, in this article, we propose a novel framework to address a new realistic problem called multiclass classification with imprecise observations (MCIMO), where we need to train a classifier with fuzzy-feature observations. First, we give the theoretical analysis of the MCIMO problem based on fuzzy Rademacher complexity. Then, two practical algorithms based on support vector machine and neural networks are constructed to solve the proposed new problem. The experiments on both synthetic and real-world datasets verify the rationality of our theoretical analysis and the efficacy of the proposed algorithms.**

***Index Terms*—Classification, fuzzy vector, machine learning.**

I. Introduction

MACHINE learning methods for the multiclass classification problem have gained great achievements in many areas, including medical imaging [1], natural language processing [2], biology [3], and computer vision [4]. The theoretical analysis of existing well-known multiclass classification machine learning algorithms, such as *support vector machine* (SVM) [5] and neural networks [6], has been well researched [7]. Recently, many researchers considered using different measures to give the estimation error bounds for classification problems that can guarantee the rationality of these algorithms. These measures include the Rademacher complexity [7]–[9], VC-dimension [10], [11], stability and probably approximately correct (PAC)-Bayesian [12], [13], and local Rademacher Complexity [14], [15].

The Rademacher complexity is a crucial tool to derive generalization bounds, which measure how well a given hypothesis set can fit random noise. A Rademacher complexity-based bound was first proposed by Koltchinskii and Panchenko [8]. Subsequently, this bound was improved in [7]. Then, Maximov *et al.* [9] presented a new estimation error bound using Rademacher complexity for multiclass classification issues. In addition, to ensure multiclass PAC learnability, a series of estimation error bounds based on VC-dimension and Natarajan dimension was proposed in [10] and [11]. Because of the dependence on dimensions, these VC-dimension-based bounds rarely apply to large-scale issues. To conduct theoretical analysis of neural networks for multiclass classification problems, Hardt *et al.* [12] and McAllester [13] introduced the new bounds based on stability and PAC-Bayesian. Furthermore, tighter and sharper bounds were proposed in [14] and [15] by using local Rademacher complexity. According to these theoretical analyses, it illustrates that we can always learn a good classifier for multiclass classification problems to predict the test set when the instances are precise in the training and test sets with the same distribution and enough instances can be collected in the training set.

However, there is one limitation with multiclass classification that the existing methods cannot handle the scenario that only imprecise observations are available. For example, the readings on many measuring devices are not exact numbers but intervals because there are only a limited number of decimals available on most of these measuring devices. Thus, this scenario has inspired us to consider a further realistic problem called multiclass classification with imprecise observations (MCIMO). With the MCIMO problem, we aim to train a classifier with high classification performance for multiclass classification problems when the features of all the instances in both training and test sets are imprecise (e.g., fuzzy-valued or interval-valued features).

The main challenge to solving the MCIMO problem is how to handle observations with fuzzy-valued or interval-valued features. The existing well-known machine learning methods cannot be directly used to address the MCIMO problem. Recently, combining fuzzy techniques with machine learning methods (especially for transfer learning methods [16]–[20]) has drawn increasing attention. In the literature review section, we will give a brief review of these machine learning methods with fuzzy techniques [21]–[26]. According to these fuzzy-based methods, it demonstrates that fuzzy techniques are powerful tools to analyze imprecise observations and

Manuscript received March 20, 2022; revised May 3, 2022; accepted June 4, 2022. This work was supported by the Australian Research Council (ARC) under Grant FL190100149. This article was recommended by Associate Editor N. Zhang. *(Corresponding author: Jie Lu.)*

The authors are with the Australian Artificial Intelligence Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: guangzhi.ma@student.uts.edu.au; jie.lu@uts.edu.au; feng.liu@uts.edu.au; zhen.fang@uts.edu.au; guangquan.zhang@uts.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2022.3181193.

Digital Object Identifier 10.1109/TCYB.2022.3181193

provide better interpretability to handle the uncertainty of different issues. Therefore, we consider using fuzzy techniques to address the MCIMO problem because they can represent the imprecise features of the instances in both training and test sets and can handle different types of uncertainty issues.

In this article, we consider using the fuzzy random variable, which was proposed in [27] and [28], to represent the imprecise feature of the instances. Then, we give the theoretical analysis and obtain the estimation error bounds for the MCIMO problem. In the MCIMO problem, these bounds are really important as it ensures that we can always train a fuzzy classifier with high classification accuracy when the instances are drawn from the same fuzzy distribution and enough fuzzy-feature instances can be collected.

Subsequently, we construct two fuzzy technique-based algorithms, which combine fuzzy techniques with SVM and neural networks to analyze fuzzy data. The proposed algorithms contain two main parts. The first part aims to extract the most significant crisp-valued information from imprecise observations, which is the main difficulty of the proposed algorithms. In this article, we compare the performance of different defuzzification methods on synthetic datasets to find the optimal defuzzification function for the proposed algorithms. The second part is to classify the extracted crisp-valued information by two well-known machine learning methods: 1) SVM and 2) neural networks. In addition, interval-valued data are also a common type of imprecise data in real-world scenarios. In this article, we give one approach to apply the proposed methods to analyze interval-valued data. Finally, experimental results on both synthetic and real-world datasets reveal the superiority of the proposed algorithms and demonstrate that the proposed fuzzy-based methods can obtain better performance to analyze fuzzy data or interval-valued data than nonfuzzy methods through comparisons with seven baselines. The main contributions of this article are as follows.

1) We identify a novel problem called MCIMO, which considers addressing the multiclass classification problem when only imprecise observations are available, and we propose a framework to handle this problem. Based on this framework, two fuzzy technique-based machine learning algorithms called defuzzified SVM (DF-SVM) and defuzzified multilayer perception (DF-MLP) are constructed, which combine fuzzy techniques with SVM and neural networks. These algorithms significantly improve classification accuracy since they use fuzzy vectors to express the distribution of imprecise data and apply different defuzzification methods to extract crisp-valued information from imprecise observations.
2) We give the theoretical analysis of the MCIMO problem based on the fuzzy Rademacher complexity, which ensures that we can always train a fuzzy classifier with high classification accuracy. This theory provides a theoretical basis for fuzzy data analysis.
3) By comparing the performance of different defuzzification methods on synthetic datasets, we find the optimal defuzzification function for the fuzzy technique-based SVM and neural networks algorithms. Through experimental comparisons with several baselines on both synthetic and real-world datasets, it demonstrates the superiority of the proposed algorithms to analysis fuzzy data and interval-valued data.

The remainder of this article is structured as follows. Section II presents a brief review of the methods, which combine fuzzy techniques with machine learning methods. Section III introduces the related definitions. Section IV introduces and gives a formal definition of the MCIMO problem. Section V gives the theoretical analysis of the MCIMO problem. Section VI proposes a novel framework to address the MCIMO problem and constructs two algorithms based on this framework to analyze fuzzy-feature observations. In Sections VII and VIII, the experiments on both synthetic and real-world datasets are constructed to show the superiority of the proposed algorithms. Section IX concludes this article and outlines future work.

II. Literature Review

In this section, a brief review of the methods, which combine fuzzy techniques with machine learning methods, is presented.

On the one hand, for classification tasks, Colubi *et al.* [21] integrated fuzzy L_2 metrics [29] with the discriminant analysis approach to analyze fuzzy data. Yang *et al.* [30] proposed a novel fuzzy SVM algorithm based on a kernel fuzzy c-means clustering method to deal with the classification problems with outliers or noises. Rong *et al.* [31] introduced a new classification method, which applies the defuzzified Choquet integral to address heterogeneous fuzzy data classification issues. Wang *et al.* [22] presented a novel deep-ensemble-level-based Takagi–Sugeno–Kang (TSK) fuzzy classifier to address imbalanced data classification tasks, which achieved both promising classification performance and high interpretability of zero-order TSK fuzzy classifiers. Liu *et al.* [32] used fuzzy vectors to model imprecise observations of distributions and help address the two-sample testing problem that is a core problem in the machine learning field [33]–[35].

In addition, in the area of transfer learning, Behbood *et al.* [36], [37] proposed a series of novel fuzzy-based transfer learning methods for long-term bank failure prediction, which use the fuzzy sets and the concepts of similarity and dissimilarity to modify the labels of the target instances. Deng *et al.* [38]–[41] proposed several new approaches that integrate TSK fuzzy system (TSK-FS) with transfer learning to recognize epileptic electroencephalogram signals. To solve the heterogeneous unsupervised domain adaptation (HeUDA) problems for classification tasks, Liu *et al.* [42] introduced a novel HeUDA approach utilizing shared fuzzy equivalence relations via fuzzy geometry, which can measure the similarity between the features of the instances in the source and target domain. Furthermore, Liu *et al.* [23] enhanced this method, which called the shared-fuzzy-equivalence-relations neural network to analyze another challenging problem called the multisource HeUDA.

In contrast, for regression tasks, Deng *et al.* [43], [44] proposed several novel transfer learning approaches utilizing the Mamdani–Larsen fuzzy systems and TSK-FS.

Furthermore, Deng *et al.* [45] improved the above model to construct a new transfer learning model that uses two knowledge-leverage strategies, learning from the TSK-FS model, to enhance the two types of parameters for the target domain. In addition, Zuo *et al.* [46] applied granular computing techniques to transfer learning and proposed a comprehensive domain adaptation framework based on the T–S fuzzy model. Subsequently, Zuo *et al.* [24] presented a novel fuzzy rule-based transfer learning model, which integrates an infinite Gaussian mixture model with active learning. Applying these two techniques, researchers can identify the data structure and select an appropriate source domain when multisource domains are available, and choose labeled data for the target model with high efficiency when the target domain contains insufficient data. Hence, Lu *et al.* [25] presented a novel fuzzy rule-based transfer learning approach that merges fuzzy rules from multisource domains in both homogeneous and heterogeneous scenarios. Besides, some new fuzzy-based clustering methods were presented in [47] and [48] to analyze fuzzy data.

In our previous work [26], we proposed one algorithm to solve a novel classification problem that the instances in training and test sets are all imprecise and we give the theoretical analysis of this problem. However, there are two drawbacks in our previous works. First, one gap has not be solved that there is no research to explore properties of different defuzzification methods. Second, we only verified the performance of the proposed algorithm on the synthetic dataset, while the performance of the proposed algorithm on real-world datasets is indispensable. In this article, we address both drawbacks in our previous work.

III. Preliminary

In this section, some related definitions are introduced, including the definitions of fuzzy probability density function and fuzzy probability distribution.

Definition 1 [28]: Let R be the universal set and $\widetilde{X}$ be a fuzzy random variable. Suppose $f_{\widetilde{X}_\alpha}(x)$ is the probability density function of $\widetilde{X}^L_\alpha$ and $\widetilde{X}^U_\alpha$, where $[\widetilde{X}^L_\alpha, \widetilde{X}^U_\alpha]$ is the α-cut of $\widetilde{X}$. We define $\widetilde{f}(\widetilde{x})$ as the fuzzy probability density function of $\widetilde{X}$. Then, the membership function of $\widetilde{f}(\widetilde{x})$ is defined as

$$\mu_{\widetilde{f}(\widetilde{x})}(r) = \sup_{0\le\alpha\le1} \alpha 1_{A_\alpha}(r) \tag{1}$$

where

$$\begin{aligned} A_\alpha &= \left[\min_{x\in[\widetilde{x}^L_\alpha,\widetilde{x}^U_\alpha]} f_{\widetilde{X}_\alpha}(x), \max_{x\in[\widetilde{x}^L_\alpha,\widetilde{x}^U_\alpha]} f_{\widetilde{X}_\alpha}(x)\right] \\ &= \left[\min\left\{\min_{\alpha\le\beta\le1} f_{\widetilde{X}_\alpha}\left(\widetilde{x}^L_\beta\right), \min_{\alpha\le\beta\le1} f_{\widetilde{X}_\alpha}\left(\widetilde{x}^U_\beta\right)\right\}\right. \\ &\qquad \left.\max\left\{\max_{\alpha\le\beta\le1} f_{\widetilde{X}_\alpha}\left(\widetilde{x}^L_\beta\right), \max_{\alpha\le\beta\le1} f_{\widetilde{X}_\alpha}\left(\widetilde{x}^U_\beta\right)\right\}\right]. \end{aligned}$$

Definition 2 [26]: We denote $\widetilde{D}$ as the fuzzy probability distribution of $\widetilde{X} \in \mathcal{F}_{\mathbb{R}}$ (denoted as $\widetilde{X} \sim \widetilde{D}$), which contains the value range and fuzzy probability density function of $\widetilde{X}$, where D represents the value range of real-valued variable x that induce all fuzzy real numbers in $\widetilde{D}$.

Let $\widetilde{X} = (\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_p) \in \mathcal{F}^p_{\mathbb{R}^p}$ be the p-fuzzy random vector, where $\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_p \in \mathcal{F}_{\mathbb{R}}$ are i.i.d fuzzy random variables. Suppose the probability density function of $\widetilde{x}_j$ is $\widetilde{f}_j(\widetilde{x})$, $j = 1, \ldots, p$. We denote the joint probability density function of $\widetilde{X}$ as $\widetilde{f}_{\widetilde{X}}(\widetilde{x}) = \widetilde{f}_1(\widetilde{x}_1) \bigotimes \cdots \bigotimes \widetilde{f}_p(\widetilde{x}_p)$ and its membership function is defined by

$$\xi_{\widetilde{f}_{\widetilde{X}}(\widetilde{x})}(r) = \sup_{0\le\alpha\le1} 1_{[\widetilde{f}_{\widetilde{X}}(\widetilde{x})]_\alpha}(r) \tag{2}$$

where

$$\begin{aligned} &\left[\widetilde{f}_{\widetilde{X}}(\widetilde{x})\right]_\alpha \\ &= \left[\prod_{j=1}^{p} \min_{x_j\in[\widetilde{x}^L_{j\alpha},\widetilde{x}^U_{j\alpha}]} f_{\widetilde{x}_{j\alpha}}(x_j), \prod_{j=1}^{p} \max_{x_j\in[\widetilde{x}^L_{j\alpha},\widetilde{x}^U_{j\alpha}]} f_{\widetilde{x}_{j\alpha}}(x_j)\right] \\ &= \left[\prod_{j=1}^{p} \min\left\{\min_{\alpha\le\beta\le1} f_{\widetilde{x}_{j\alpha}}\left(\widetilde{x}^L_{j\beta}\right), \min_{\alpha\le\beta\le1} f_{\widetilde{x}_{j\alpha}}\left(\widetilde{x}^U_{j\beta}\right)\right\}\right. \\ &\qquad \left.\prod_{j=1}^{p} \max\left\{\max_{\alpha\le\beta\le1} f_{\widetilde{x}_{j\alpha}}\left(\widetilde{x}^L_{j\beta}\right), \max_{\alpha\le\beta\le1} f_{\widetilde{x}_{j\alpha}}\left(\widetilde{x}^U_{j\beta}\right)\right\}\right]. \end{aligned}$$

Then, we denote $\widetilde{\mathcal{D}}$ as the fuzzy distribution over $\widetilde{\mathcal{X}} \subset \mathcal{F}^p_{\mathbb{R}^p}$, where $\widetilde{\mathcal{D}}$ contains the value range and the joint probability density function of any fuzzy vector belongs to $\widetilde{\mathcal{X}}$.

IV. Multiclass Classification With Imprecise Observations

In this section, we introduce the MCIMO problem. Let $\widetilde{\mathcal{X}} \subset \mathcal{F}^p_{\mathbb{R}^p}$ be the input space and $\mathcal{Y} = [1, K]$ be the output space, and let $\widetilde{\mathcal{D}}$ be an unknown fuzzy distribution over $\widetilde{\mathcal{X}}$. Suppose $\widetilde{S} = \{(\widetilde{X}_i, y_i)\}^m_{i=1}$ be a sample drawn from $\widetilde{\mathcal{X}} \times \mathcal{Y}$, where $\widetilde{X}_i = (\widetilde{x}_{i1}, \widetilde{x}_{i2}, \ldots, \widetilde{x}_{ip})$, $i = 1, 2, \ldots, m$ drawn i.i.d. from $\widetilde{\mathcal{D}}$ and $y_i = f(\widetilde{X}_i)$ is the ground-truth function denoted as

$$\begin{aligned} f : \widetilde{\mathcal{X}} &\to \mathcal{Y} \\ \left(\widetilde{x}_{i1}, \widetilde{x}_{i2}, \ldots, \widetilde{x}_{ip}\right) &\to k. \end{aligned}$$

We noticed that if $\widetilde{X}_i \in \widetilde{\mathcal{X}}$ belongs to the kth class, then $f(\widetilde{X}_i) = k$. Let $\mathcal{H} \subset \{h : \widetilde{\mathcal{X}} \to \mathbb{R}^K\}$ be the hypothesis set of the MCIMO problem and $\forall h \in \mathcal{H}$

$$\begin{aligned} h : \widetilde{\mathcal{X}} &\to \mathbb{R}^K \\ \left(\widetilde{x}_{i1}, \ldots, \widetilde{x}_{ip}\right) &\to \left(h_1\left(\widetilde{X}_i\right), \ldots, h_K\left(\widetilde{X}_i\right)\right) \end{aligned}$$

where each $h_k(\widetilde{X}_i)$, $k = 1, \ldots, K$ represents the probability of the instance $\widetilde{X}_i$ belongs to the kth category. Then, we give the definition of the loss function with respect to h

$$l : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}_+.$$

Let $L_{\mathcal{H}} = \{l(h(\widetilde{X}), y) | \widetilde{X} \in \widetilde{\mathcal{X}}, h \in \mathcal{H}, y \in \mathcal{Y}\}$ be the class of loss functions associated with $\mathcal{H}$.

The traditional multiclass classification problems aim to use the sample $\widetilde{S}$ to find a hypothesis $h \in \mathcal{H}$, which can cause as small as possible risk $R(h)$ with respect to f. In the MCIMO problem, the purpose is similar to traditional multiclass classification problems. Then, we give the definition of the risk with respect to h

$$R_{\widetilde{\mathcal{D}}}(h) \triangleq R\left(l\left(h\left(\widetilde{X}\right), y\right)\right) = E_{\widetilde{X}\sim\widetilde{\mathcal{D}}}\left[l\left(h\left(\widetilde{X}\right), y\right)\right] \tag{3}$$

where the notion of $E_{\widetilde{X}\sim\widetilde{\mathcal{D}}}[l(h(\widetilde{X}), y)]$ can be found in [26].

Thus, to address the MCIMO problem, we are committed to find the optimal hypothesis function h^* to minimize the risk, that is, $h^* = \arg\min_{h \in \mathcal{H}} R_{\widetilde{\mathcal{D}}}(h)$.

V. Theoretical Analysis of the MCIMO Problem

In this section, the theoretical analysis of the MCIMO problem is presented. First, the notion of fuzzy Rademacher complexity is introduced. Then, we obtain the estimation error bounds of the MCIMO problem, which guarantees that we can always obtain a fuzzy classifier with high classification accuracy when infinite fuzzy-feature instances are available.

Definition 3 [26]: Let $L_{\mathcal{H}}$ be a family of loss functions and $\widetilde{S} = \{(\widetilde{X}_i, y_i)\}_{i=1}^m$ be a sample drawn from $\mathcal{F}_{\mathbb{R}^p}^p \times \mathcal{Y}$. Then, the empirical fuzzy Rademacher complexity of $L_{\mathcal{H}}$ and $\mathcal{H}$ with respect to the sample $\widetilde{S}$ and $\widetilde{S}_X = \{\widetilde{X}_i\}_{i=1}^m$ is defined as

$$\widehat{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{H}}) = E_{\vec{\sigma}}\left[\sup_{l \in L_{\mathcal{H}}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i l\big(h\big(\widetilde{X}_i\big), y_i\big)\right]$$
$$\widehat{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{H}) = E_{\vec{\sigma}}\left[\sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \sigma_{ik} h_k\big(\widetilde{X}_i\big)\right] \tag{4}$$

where $\vec{\sigma} = (\sigma_1, \ldots, \sigma_m)^T$, with σ_is independent random variables drawn from the Rademacher distribution, that is, $\Pr(\sigma_i = +1) = \Pr(\sigma_i = -1) = (1/2), i = 1, \ldots, m$.

Definition 4 [26]: Let $\widetilde{\mathcal{D}}' \triangleq \widetilde{\mathcal{D}} \times \mathcal{Y}$ and $\widetilde{\mathcal{D}}$ denote the fuzzy distribution according to $\widetilde{S}$ and $\widetilde{S}_X$. Then, the fuzzy Rademacher complexity of $L_{\mathcal{H}}$ and $\mathcal{H}$ is defined as follows:

$$\widetilde{\mathcal{R}}_{\widetilde{S} \sim \widetilde{\mathcal{D}}'}(L_{\mathcal{H}}) = E_{\widetilde{\mathcal{D}}'}\big[\widehat{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{H}})\big]$$
$$\widetilde{\mathcal{R}}_{\widetilde{S}_X \sim \widetilde{\mathcal{D}}}(\mathcal{H}) = E_{\widetilde{\mathcal{D}}}\big[\widehat{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{H})\big]. \tag{5}$$

Using related lemmas and theorems (shown in [26]) and the theoretical analysis of traditional multiclass classification algorithms (shown in [7]–[10] and [15]), the estimation error bounds with hypotheses $\mathcal{H}\}$ are shown in the following theorem.

Theorem 1 [26]: Let $\widetilde{S} = \{(\widetilde{X}_i, y_i)\}_{i=1}^m$ and $\widetilde{S}_X = \{\widetilde{X}_i\}_{i=1}^m, \widetilde{X}_i \sim \widetilde{\mathcal{D}} \in \widetilde{\mathcal{X}}, y_i = f(\widetilde{X}_i)$, and suppose that there are $C_l, C_h > 0$ such that $\sup_{h \in \mathcal{H}} \|h\|_\infty \leq C_h$ and $\sup_{\|h\|_\infty \leq C_h} \max_y l(t, y) \leq C_l$, and $\forall l \in L_{\mathcal{H}}$ is L_l-Lipschitz functions. For any $\delta > 0$, with fuzzy probability at least $1 - \delta$, each of the following holds for all $l \in L_{\mathcal{H}}$:

$$\left|E_{\widetilde{X} \sim \widetilde{\mathcal{D}}}\big[l\big(h\big(\widetilde{X}\big), y\big)\big] - \frac{1}{m} \sum_{i=1}^{m} l\big(h\big(\widetilde{X}_i\big), y_i\big)\right|$$
$$\leq 2\widetilde{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{H}}) + C_l\sqrt{\frac{2\log(1/\delta)}{m}}$$
$$\left|E_{\widetilde{X} \sim \widetilde{\mathcal{D}}}\big[l\big(h\big(\widetilde{X}\big), y\big)\big] - \frac{1}{m} \sum_{i=1}^{m} l\big(h\big(\widetilde{X}_i\big), y_i\big)\right|$$
$$\leq 2\widehat{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{H}}) + 3C_l\sqrt{\frac{2\log(2/\delta)}{m}}. \tag{6}$$

Because $\forall l \in L_{\mathcal{H}}$ is L_l-Lipschitz functions, we have

$$\widehat{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{H}}) \leq \sqrt{2}L_l\widehat{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{H})$$
$$\widetilde{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{H}}) \leq \sqrt{2}L_l\widetilde{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{H}). \tag{7}$$

Then

$$\big|R_{\widetilde{\mathcal{D}}}(h) - \widehat{R}_{\widetilde{\mathcal{D}}}(h)\big| \leq 2\sqrt{2}L_l\widetilde{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{H}) + C_l\sqrt{\frac{2\log(1/\delta)}{m}}$$
$$\big|R_{\widetilde{\mathcal{D}}}(h) - \widehat{R}_{\widetilde{\mathcal{D}}}(h)\big| \leq 2\sqrt{2}L_l\widehat{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{H}) + 3C_l\sqrt{\frac{2\log(2/\delta)}{m}}. \tag{8}$$

The detailed proof of Theorem 1 can be found in [26].

In Section VI, we decompose the hypothesis function into the defuzzification function and optimization function. We let the loss function $l(h(\widetilde{X}_i), y_i) = l(g(M(\widetilde{X}_i)), y_i)$, where g is a optimization function that maps $\mathbb{R}^p$ into $\mathbb{R}^K$. Let $\mathcal{M} \subset \{M : \widetilde{\mathcal{X}} \to \mathbb{R}^p\}$ be the class of defuzzification functions, $\mathcal{G}_{\mathcal{M}} \subset \{g(M(\widetilde{X})) : \mathbb{R}^p \to \mathbb{R}^K | M \in \mathcal{M}, y \in \mathcal{Y}\}$ be the class of optimization functions associated with $\mathcal{M}$, and $L_{\mathcal{G}} = \{l(g(M(\widetilde{X}_i)), y) | M \in \mathcal{M}, g \in \mathcal{G}, y \in \mathcal{Y}\}$ be the class of loss functions associated with $\mathcal{G}$. Then, we have

$$\widehat{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{G}}) = E_{\vec{\sigma}}\left[\sup_{l \in L_{\mathcal{G}}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i l\big(g\big(M\big(\widetilde{X}_i\big)\big), y_i\big)\right]$$
$$\widehat{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{G}_{\mathcal{M}}) = E_{\vec{\sigma}}\left[\sup_{g \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \sigma_{ik} g_k\big(M\big(\widetilde{X}_i\big)\big)\right]$$
$$\widehat{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{M}) = E_{\vec{\sigma}}\left[\sup_{M \in \mathcal{M}} \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} \sum_{j=1}^{p} \sigma_{ikj} M\big(\widetilde{x}_{ij}\big)\right]. \tag{9}$$

Then, we can obtain the following theorem using theorem 1.

Theorem 2 [26]: Let $\widetilde{S} = \{(\widetilde{X}_i, y_i)\}_{i=1}^m$ and $\widetilde{S}_X = \{\widetilde{X}_i\}_{i=1}^m, \widetilde{X}_i \sim \widetilde{\mathcal{D}} \in \widetilde{\mathcal{X}}, y_i = f(\widetilde{X}_i)$, and suppose that there are $C, C_l > 0$ such that $\sup_{g \in \mathcal{G}} \|g\|_\infty \leq C$ and $\sup_{\|g\|_\infty \leq C} \max_y l(t, y) \leq C_l$, and $\forall l \in L_{\mathcal{G}}$ is L_l-Lipschitz functions. For any $\delta > 0$, with fuzzy probability at least $1 - \delta$, each of the following holds for all $g \in L_{\mathcal{G}}$:

$$\left|E_{\widetilde{X} \sim \widetilde{\mathcal{D}}}\big[l\big(g(M(\widetilde{X})), y\big)\big] - \frac{1}{m} \sum_{i=1}^{m} l\big(g(M(\widetilde{X}_i)), y_i\big)\right|$$
$$\leq 2\widetilde{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{G}}) + C_l\sqrt{\frac{2\log(1/\delta)}{m}}$$
$$\left|E_{\widetilde{X} \sim \widetilde{\mathcal{D}}}\big[l(g(M(\widetilde{X})), y)\big] - \frac{1}{m} \sum_{i=1}^{m} l\big(g(M(\widetilde{X}_i)), y_i\big)\right|$$
$$\leq 2\widehat{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{G}}) + 3C_l\sqrt{\frac{2\log(2/\delta)}{m}}. \tag{10}$$

Because $\forall l \in L_{\mathcal{G}}$ is L_l-Lipschitz functions, we have

$$\widehat{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{G}}) \leq \sqrt{2}L_l\widehat{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{G}_{\mathcal{M}})$$
$$\widetilde{\mathcal{R}}_{\widetilde{S}}(L_{\mathcal{G}}) \leq \sqrt{2}L_l\widetilde{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{G}_{\mathcal{M}}). \tag{11}$$

Then

$$\big|R_{\widetilde{\mathcal{D}}}(h) - \widehat{R}_{\widetilde{\mathcal{D}}}(h)\big| \leq 2\sqrt{2}L_l\widetilde{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{G}_{\mathcal{M}}) + C_l\sqrt{\frac{2\log(1/\delta)}{m}}$$
$$\big|R_{\widetilde{\mathcal{D}}}(h) - \widehat{R}_{\widetilde{\mathcal{D}}}(h)\big| \leq 2\sqrt{2}L_l\widehat{\mathcal{R}}_{\widetilde{S}_X}(\mathcal{G}_{\mathcal{M}}) + 3C_l\sqrt{\frac{2\log(2/\delta)}{m}}. \tag{12}$$

The proof of Theorem 3 is similar to Theorem 1.

Next, we consider the estimation error bounds for kernel-based optimization functions such as SVM. Let $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a PDS kernel function, $\Phi : \mathbb{R}^p \to \mathbb{H}$ be a feature mapping associated to K, and $w_1, \ldots, w_K \in \mathbb{H}$ are weight vectors. For any $p \geq 1$, the family of kernel-based hypotheses is denoted as

$$\mathcal{G}_{K,p} = \Big\{ g : M(\widetilde{X}) \to \big(w_1^T \Phi(M(\widetilde{X})), \ldots, w_K^T \Phi(M(\widetilde{X}))\big) \\ W = \big(w_1^T, \ldots, w_K^T\big)^T, ||W||_{\mathbb{H},p} \leq \Lambda \Big\}$$

where $||W||_{\mathbb{H},p} = (\sum_{l=1}^{K} ||w_l||_{\mathbb{H}}^p)^{1/p}$. Hence, the fuzzy Rademacher complexity of $\mathcal{G}_{K,p}$ can be bounded as follows.

Lemma 1: Let $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a PDS kernel function and $\Phi : \mathbb{R}^p \to \mathbb{H}$ be a feature mapping associated to K. Assume that there exists $r > 0$ such that $K(M(\widetilde{X}), M(\widetilde{X})) \leq r^2$ for all $\widetilde{X} \in \widetilde{\mathcal{X}}$. Let $\widetilde{S}_X = \{\widetilde{X}_i\}_{i=1}^m, \widetilde{X}_i \sim \widetilde{\mathcal{D}} \in \widetilde{\mathcal{X}}$. Then, for any $m \geq 1$

$$\widetilde{\mathcal{R}}_{\widetilde{S}_X \sim \widetilde{\mathcal{D}}}(\mathcal{G}_{K,p}) \leq K\sqrt{\frac{r^2\Lambda^2}{m}}. \tag{13}$$

Proof: For all $l \in [1, K]$, $||w_l||_{\mathbb{H}} \leq (\sum_{l=1}^{K} ||w_l||_{\mathbb{H}}^p)^{1/p} = ||W||_{\mathbb{H},p}$ holds. Thus, as $||W||_{\mathbb{H},p} \leq \Lambda$, we have $||w_l||_{\mathbb{H}} \leq \Lambda$ for all $l \in [1, K]$. Then, the fuzzy Rademacher complexity of the hypothesis set $\mathcal{G}_{K,p}$ can be bounded as follows:

$$\begin{aligned}
&\widetilde{\mathcal{R}}_{\widetilde{S}_X \sim \widetilde{\mathcal{D}}}(\mathcal{G}_{K,p}) \\
&= \frac{1}{m} E_{\widetilde{\mathcal{D}},\vec{\sigma}}\left[\sup_{||W|| \leq \Lambda} \sum_{i=1}^{m}\sum_{k=1}^{K} \sigma_{ik} g_k\big(M(\widetilde{X}_i)\big)\right] \\
&= \frac{1}{m} E_{\widetilde{\mathcal{D}},\vec{\sigma}}\left[\sup_{||W|| \leq \Lambda} \sum_{i=1}^{m}\sum_{k=1}^{K} \sigma_{ik} w_k^T \Phi\big(M(\widetilde{X}_i)\big)\right] \\
&\leq \frac{K}{m} E_{\widetilde{\mathcal{D}},\vec{\sigma}}\left[\sup_{k \in [K], ||W|| \leq \Lambda} \left\langle w_k, \sum_{i=1}^{m} \sigma_{ik} \Phi\big(M(\widetilde{X}_i)\big)\right\rangle\right] \\
&\quad \text{(using Cauchy-Schwarz inequality)} \\
&\leq \frac{K}{m} E_{\widetilde{\mathcal{D}},\vec{\sigma}}\left[\sup_{k \in [K], ||W|| \leq \Lambda} ||w_k||_{\mathbb{H}} \left\| \sum_{i=1}^{m} \sigma_{ik} \Phi\big(M(\widetilde{X}_i)\big)\right\|_{\mathbb{H}}\right] \\
&\leq \frac{K\Lambda}{m} E_{\widetilde{\mathcal{D}},\vec{\sigma}}\left[\sup_{k \in [K]} \left\| \sum_{i=1}^{m} \sigma_{ik} \Phi\big(M(\widetilde{X}_i)\big)\right\|_{\mathbb{H}}\right] \\
&\quad \text{(using Jensen's inequality)} \\
&\leq \frac{K\Lambda}{m}\left[E_{\widetilde{\mathcal{D}},\vec{\sigma}}\left[\sup_{k \in [K]} \left\| \sum_{i=1}^{m} \sigma_{ik} \Phi\big(M(\widetilde{X}_i)\big)\right\|_{\mathbb{H}}^2\right]\right]^{1/2} \\
&\quad \big(i \neq j \Rightarrow E_{\vec{\sigma}}\big[\sigma_{ik}\sigma_{jk}\big] = 0\big) \\
&= \frac{K\Lambda}{m}\left[E_{\widetilde{\mathcal{D}}}\left[\sum_{i=1}^{m} \|\Phi\big(M(\widetilde{X}_i)\big)\|_{\mathbb{H}}^2\right]\right]^{1/2} \\
&= \frac{K\Lambda}{m}\left[E_{\widetilde{\mathcal{D}}}\left[\sum_{i=1}^{m} K\big(M(\widetilde{X}_i), M(\widetilde{X}_i)\big)\right]\right]^{1/2} \\
&\leq K\sqrt{\frac{r^2\Lambda^2}{m}}
\end{aligned}$$

which yields the result. ■

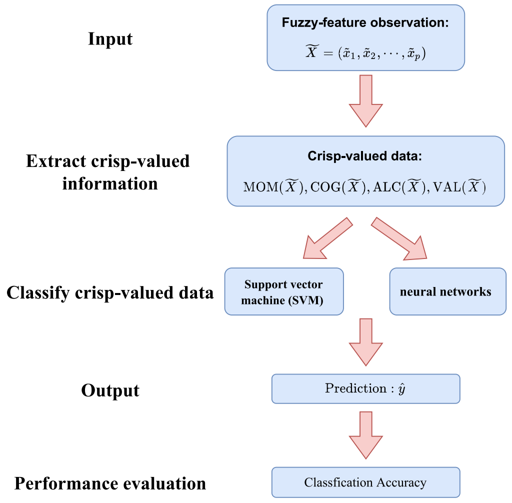

Input
Fuzzy-feature observation:
$\widetilde{X} = (\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_p)$
Extract crisp-valued information
Crisp-valued data:
$\mathrm{MOM}(\widetilde{X}), \mathrm{COG}(\widetilde{X}), \mathrm{ALC}(\widetilde{X}), \mathrm{VAL}(\widetilde{X})$
Classify crisp-valued data
Support vector machine (SVM)
neural networks
Output
Prediction : $\hat{y}$
Performance evaluation
Classfication Accuracy

Fig. 1. Framework of the proposed algorithms.

Next, combining Theorem 2 and Lemma 1 directly yields the following generalization bound.

Theorem 3: Let $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a PDS kernel function and $\Phi : \mathbb{R}^p \to \mathbb{H}$ be a feature mapping associated to K. Assume that there exists $r > 0$ such that $K(M(\widetilde{X}), M(\widetilde{X})) \leq r^2$ for all $\widetilde{X} \in \widetilde{\mathcal{X}}$. Let $\widetilde{S}_X = \{\widetilde{X}_i\}_{i=1}^m, \widetilde{X}_i \sim \widetilde{\mathcal{D}} \in \widetilde{\mathcal{X}}$ and suppose that there are $C, C_l > 0$ such that $\sup_{g \in \mathcal{G}_{K,p}} \|g\|_\infty \leq C$ and $\sup_{\|g\|_\infty \leq C} \max_y l(t, y) \leq C_l$, and $\forall l \in L_{\mathcal{G}_{K,p}}$ is L_l-Lipschitz functions. For any $\delta > 0$, with fuzzy probability at least $1-\delta$, each of the following holds for all $h \in \mathcal{G}_{K,p}$:

$$\left|R_{\widetilde{\mathcal{D}}}(h) - \widehat{R}_{\widetilde{\mathcal{D}}}(h)\right| \leq 2KL_l\sqrt{\frac{2r^2\Lambda^2}{m}} + C_l\sqrt{\frac{2\log(1/\delta)}{m}}. \tag{14}$$

According to (8), (12), and (14), we notice that fix some constants, as $m \to \infty$, $R_{\widetilde{\mathcal{D}}}(h) \to \widehat{R}_{\widetilde{\mathcal{D}}}(h)$. Therefore, these bounds demonstrate that we can always obtain a fuzzy classifier with high classification accuracy when enough fuzzy-feature instances can be collected. These theoretical analyses reveal that fuzzy classifiers can be constructed to effectively and accurately handle the MCIMO problem.

VI. Construct Fuzzy Classifiers for Solving MCIMO Problem

In this section, two fuzzy classifiers are constructed to handle the MCIMO problem. The framework of the proposed algorithms is shown in Fig. 1. In the MCIMO problem, we aim to train a fuzzy classifier for fuzzy-feature input prediction. Let $\widetilde{X}_i = (\widetilde{x}_{i1}, \widetilde{x}_{i2}, \ldots, \widetilde{x}_{ip}), i = 1, \ldots, m$ be a fuzzy-feature input, where $\widetilde{x}_{ij}, i = 1, \ldots, m, j = 1, \ldots, p$ are the fuzzy number. The commonly used fuzzy numbers include Gaussian fuzzy numbers, trapezoidal fuzzy numbers, and triangular fuzzy numbers. First, a Gaussian fuzzy number $\widetilde{x}$ can be characterized by (c, δ) and the membership function is given in the following equation:

$$\mu_{\widetilde{x}}(t) = \exp(-(t-c)/2\delta)^2.$$

A trapezoidal fuzzy number $\widetilde{x}$ can be characterized by (a_1, b_1, b_2, a_2) and the membership function of a trapezoidal fuzzy number $\widetilde{x}$ is shown as follows:

$$\mu_{\widetilde{x}}(t) = \begin{cases} 0, & t < a_1 \\ \frac{t-a_1}{b_1-a_1}, & a_1 \leq t < b_1 \\ 1, & b_1 \leq t < b_2 \\ \frac{t-a_2}{b_2-a_2}, & b_2 \leq t < a_2 \\ 0, & t \geq a_2. \end{cases}$$

Finally, when $b_1 = b_2$, a trapezoidal fuzzy number is become a triangular fuzzy number. Thus, a triangular fuzzy number $\widetilde{x}$ can be characterized by (a_1, b_1, a_2).

To address the MCIMO problem, we need to construct a hypothesis function $h \in \mathcal{H}$, which mapping the input space $\widetilde{\mathcal{X}} \subset \mathcal{F}^p_{\mathbb{R}^p}$ into $\mathbb{R}^K$. A hypothesis function h can be decomposed into a composition of two functions. The first function M, called the defuzzification function, is defined as follows:

$$\begin{aligned} M : \widetilde{\mathcal{X}} &\rightarrow \mathbb{R}^p \\ \big(\widetilde{x}_{i1}, \widetilde{x}_{i2}, \ldots, \widetilde{x}_{ip}\big) &\rightarrow \big(M(\widetilde{x}_{i1}), \ldots, M(\widetilde{x}_{ip})\big). \end{aligned}$$

Next, four different defuzzification methods are introduced.

1) The first method is called *mean/middle of maxima* (MOM) [49], which is widely used due to its calculation simplicity. MOM is defined as

$$\text{MOM}(\widetilde{x}) = \text{Mean}\Big(t = \arg\max_t \mu_{\widetilde{x}}(t)\Big). \tag{15}$$

2) *Centre of Gravity* (COG) [50] is another widely used defuzzification method. The definitions of COG for discrete and continuous situations are shown as follows:

$$\begin{aligned} \text{COG}(\widetilde{x}) &= \frac{\sum t\mu_{\widetilde{x}}(t)}{\sum \mu_{\widetilde{x}}(t)} (\text{discrete}) & (16) \\ &= \frac{\int t\mu_{\widetilde{x}}(t)dt}{\int \mu_{\widetilde{x}}(t)dt} (\text{continuous}). & (17) \end{aligned}$$

3) The third approach, called *averaging level cuts* (ALC) [51], is defined as the flat averaging of all midpoints of the α-cuts. ALC is defined as

$$\text{ALC}(\widetilde{x}) = \frac{1}{2}\int_0^1 \big(\widetilde{x}^L_\alpha + \widetilde{x}^U_\alpha\big)d\alpha. \tag{18}$$

4) The final method is called *value of a fuzzy number* (VAL) [52], which uses α-levels as weighting factors in averaging the α-cut midpoints. VAL is defined as

$$\text{VAL}(\widetilde{x}) = \int_0^1 \alpha\big(\widetilde{x}^L_\alpha + \widetilde{x}^U_\alpha\big)d\alpha. \tag{19}$$

In Section VII, we compare the performance of different defuzzification methods on synthetic datasets. The experimental results illustrate that VAL outperforms than other three defuzzification methods. Therefore, (19) is used as the defuzzification function in all subsequent experiments.

Through the first progress, the initial issue becomes a traditional multiclass classification problem with crisp data. Therefore, the second function, called the optimization function, is a hypothesis function that maps $\mathbb{R}^p$ into $\mathbb{R}^K$ to solve the traditional multiclass classification problem. Since SVM and neural networks have gained great achievements on multiclassification problems, we decide to apply both algorithms as the optimization method. Next, we will introduce both algorithms for multiclassification problems.

Algorithm 1 DF-SVM

```
1: Input training data $D_{tr}$, selected appropriate regularization parameter $C$ and kernel function ;
2: Initial Preprocessing the training data $D_{tr}$;
3: Defuzzification Using equation (19) to transform $\widetilde{D}_x = (\widetilde{X}_1, \cdots, \widetilde{X}_N)$ into $D_x = (X_1, \cdots, X_N)$;
4: Optimization
Solving $K$ optimization problems in (20);
5: Output $\overrightarrow{\alpha}^*_l = (\alpha^*_{1l}, \cdots, \alpha^*_{Nl})^T, l = 1, 2, \cdots, K$ and the decision function in (22).
```

A. Defuzzified Support Vector Machine

First, SVM (one-versus-rest SVM [53]) with the PDS kernel function is used as the optimization function to solve the MCIMO problem. Suppose $D_{tr} = ((\widetilde{X}_1, y_1), \ldots, (\widetilde{X}_N, y_N))$ is the training data, where $\widetilde{X}_i \in \widetilde{\mathcal{X}} \subset \mathcal{F}^p_{\mathbb{R}^p}, y_i \in \{-l, +l\}, l = 1, 2, \ldots, K, i = 1, 2, \ldots, N$. The $-l$ indicates that $\widetilde{X}_i$ does not belong to category l, and $+l$ represents that $\widetilde{X}_i$ belongs to category l. In the first step, defuzzification function (19) is used to transform fuzzy input $\widetilde{D}_x = (\widetilde{X}_1, \ldots, \widetilde{X}_N)$ to crisp input denoted as $D_x = (X_1, \ldots, X_N)$. Let $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a PDS kernel function. Hence, we need to solve K optimization problems separately, and the lth problem is shown as follows:

$$\begin{aligned} \min_\alpha \quad & \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N \alpha_{il}\alpha_{jl}y_iy_jK(X_i, X_j) - \sum_{i=1}^N \alpha_{il} \\ \text{s.t} \quad & \sum_{i=1}^N \alpha_{il}y_i = 0 \\ & 0 \leq \alpha_{il} \leq C, i = 1, 2, \ldots, N. \end{aligned} \tag{20}$$

The optimal solution is $\overrightarrow{\alpha}^*_l = (\alpha^*_{1l}, \ldots, \alpha^*_{Nl})^T, l = 1, 2, \ldots, K$. Then, choose a positive component $0 \leq \alpha^*_{jl} \leq C$ of $\overrightarrow{\alpha}^*_l$, and calculate

$$b^*_l = y_j - \sum_{i=1}^N \alpha^*_{il}y_iK(X_i, X_j). \tag{21}$$

Finally, the decision function is given as follows:

$$h(X) = \arg\max_{l\in[K]}\left(\sum_{i=1}^N \alpha^*_{il}y_iK(X, X_i) + b^*_l\right). \tag{22}$$

The following algorithm called DF-SVM is shown in Algorithm 1.

B. Defuzzified Multilayer Perception

Second, a multilayer perception model, which contains two hidden layers and an output layer (softmax), is used as the optimization function to complete the second progress. We denote the parameters of the two hidden layers are W_1, b_1 and W_2, b_2, respectively, and the parameters of the output layer are W_0, b_0, respectively, and the activation function is ϕ. Then, the

Algorithm 2 DF-MLP [26]

```
1: Input training data $D_{tr}$, learning rate $\eta$, fixed epoch $T_{max}$, loss function (cross-entropy loss function is selected) and optimization algorithm (Adam algorithm [54] is selected);
2: Initial $W_0^0, W_1^0, W_2^0, b_0^0, b_1^0, b_2^0$;
for $T = 1, 2, \ldots, T_{max}$ do
    3: Fetch mini-batch $\check{D}_{tr}$ from $D_{tr}$;
    4: Calculate
    $L = loss(h(\widetilde{X}; W_0^{T-1}, W_1^{T-1}, W_2^{T-1}, b_0^{T-1}, b_1^{T-1}, b_2^{T-1}), \widehat{y})$
    according to Eqs. (19) and (23);
    5: Update $W_0^T, W_1^T, W_2^T, b_0^T, b_1^T, b_2^T = \text{Adam}(L)$;
end
6: Output $W_0^{T_{max}}, W_1^{T_{max}}, W_2^{T_{max}}, b_0^{T_{max}}, b_1^{T_{max}}, b_2^{T_{max}}$.
```

outcome of the constructed multilayer perception model can be expressed as when we obtain a fuzzy-feature input $\widetilde{X}$

$$O(\widetilde{X}) = \phi\big(\phi\big(M(\widetilde{X})W_1 + b_1\big)W_2 + b_2\big)W_0 + b_0$$
$$\widehat{y} = \arg\max_{k\in\{1,2,\ldots,K\}}\big(h_k(\widetilde{X})\big) \tag{23}$$

where

$$h(\widetilde{X}) = \big(h_1(\widetilde{X}), \ldots, h_K(\widetilde{X})\big) = \text{softmax}\big(O(\widetilde{X})\big).$$

The following algorithm called DF-MLP is shown in Algorithm 2.

VII. Experiments on Synthetic Datasets

In this section, we first compare the performance of different defuzzification methods on synthetic datasets to select the optimal defuzzification function for the proposed algorithms. Then, we verify the efficacy of the proposed algorithms for solving the MCIMO problem by comparing seven baselines in terms of classification accuracy on synthetic datasets.

A. Dataset Generation

In this section, we introduce how to construct the synthetic dataset (balanced data), which contains N fuzzy instances distributed in five categories. Each instance has 20 fuzzy features. First, we generate the real-valued vectors $X_i = (x_{i1}, \ldots, x_{i20}), i = 1, \ldots, N$ in five categories by a random number generator as the true value of the instance. Then, we use the generated real-valued vectors to construct the observation datasets $\{\widetilde{X}_i = (\widetilde{x}_{i1}, \ldots, \widetilde{x}_{i20})\}_{i=1}^N$. Each $\widetilde{x}_{ij}$ is a triangular fuzzy number characterized by $(x_{ij}-a_{ij}, x_{ij}+b_{ij}, x_{ij}+c_{ij})$ where $a_{ij} \sim U[1.5, 3], b_{ij} \sim U[-0.5, 0.5], c_{ij} \sim U[2, 4]$ and $U[a, b]$ denotes the uniform distribution over $[a, b]$.

B. Experimental Setup

In this section, baselines and experimental details of all baselines, DF-SVM and DF-MLP, are introduced.

1) Baselines: First, we introduce the first five baselines, which called Meanlogistic, MeanSVM, MeanDecisiontree, MeanRandomForest, and MeanMLP. For the fuzzy-feature dataset, a fuzzy feature is denoted as $\widetilde{x} = (\inf P_0, \sup P_0, \inf P_1, \sup P_1)$. We use $M_1(\widetilde{x}) = (\inf P_0 + \sup P_0 + \inf P_1 + \sup P_1)/4$ to transfer fuzzy features to crisp features. For interval-valued datasets, $x = [A, B]$ is denoted as an interval-valued feature. Similarly, $M_2(x) = (A + B)/2$ is used to transfer interval-valued features to crisp features. Then, those baselines apply five well-known machine learning methods (logistic regression, SVM, decision trees, random forests, and neural networks) to classify crisp-valued data obtained with the above-mentioned methods. Second, the last two baselines called DCCF and BCCF are presented in [21].

2) Experimental Details: For DF-MLP, we let momentum = 0.9 and weight decay = 0.0001. Finally, for the DCCF and BCCF algorithms, φ is selected to be the Lebesgue measure on $[0, 1]$ and $\theta = 1/3$, $K(u) = (15/8)(1 - u^2)^2 I_{(u\in[0,1])}$ is used as the kernel function. All these settings of DCCF and BCCF algorithms can obtain the best performance from [21]. However, DCCF and BCCF algorithms can only process the fuzzy data with one fuzzy feature, whereas the generated synthetic datasets contain multiple fuzzy features. Therefore, we consider using the average distance between each fuzzy feature to represent the distance between the fuzzy feature vectors in the DCCF and BCCF algorithms.

For each algorithm on each dataset, we randomly divide each dataset into the training set, the validation set, and the test set, which contain 60%, 20%, and 20% of the data, respectively. First, we select the hyperparameters that can obtain the highest average classification accuracy on the validation set. The average classification accuracy on the validation set is the average of the results of ten repeated experiments on the validation set. The hyperparameters that need to be selected are shown in Table I. Then, the selected optimal hyperparameters are used to test the performance of each algorithm on the test set. We repeat the entire experiment process 20 times. Thus, the final results are shown in the form of "mean± standard deviation." To avoid random errors, we randomly scramble the data before each experiment. Classification accuracy is used to evaluate the performance of the proposed model. The definition of classification accuracy is shown as follows:

$$\text{Accuracy} = \frac{\big|\widetilde{X} \in \widetilde{\mathcal{X}} : f(\widetilde{X}) = h(\widetilde{X})\big|}{\big|\widetilde{X} \in \widetilde{\mathcal{X}}\big|}$$

where $f(\widetilde{X})$ is the ground-truth label of $\widetilde{X}$, while $h(\widetilde{X})$ is the label predicted by the presented algorithms and the baselines.

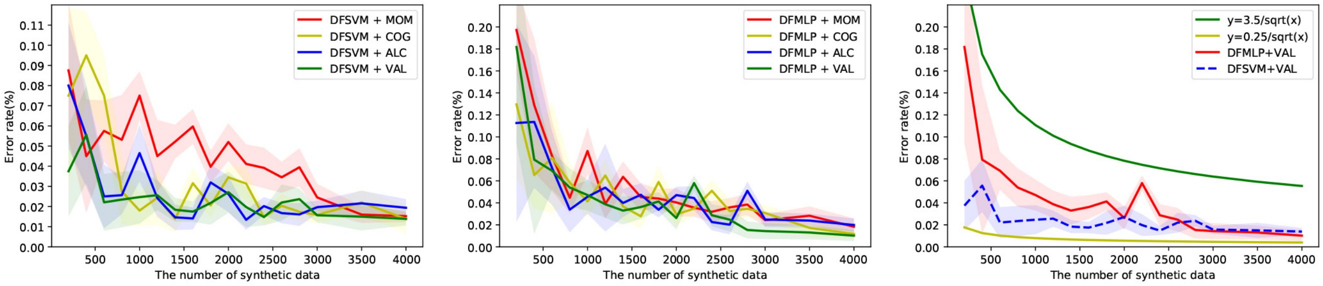
In the first experiment, we compare the performance of the proposed two algorithms with different defuzzification functions on the test set when the number of synthetic data increases. The number of synthetic data N is selected from $\{200, 400, \ldots, 3000, 3500, 4000\}$. In the second experiment, we generated 2000 synthetic data and analyzed them using the proposed methods and baselines, respectively. In addition, the Wilcoxon rank-sum test results of the method, which obtains the best performance, with other methods are given.

C. Experimental Results Analysis

The results of the first experiment are shown in Fig. 2. From Figs. 2(a) and (b), we find that COG and VAL have better performance than another two methods in terms of convergence speed and classification error and VAL is more stable

TABLE I
HYPERPARAMETERS FOR THE PROPOSED ALGORITHMS AND SEVEN BASELINES

Algorithm	Hyperparameters	Ranges
Meanlogistic	regularization parameter C	$\{0.1, 0.2, \cdots, 0.9, 1, 2, \cdots, 100\}$
MeanSVM	regularization parameter C, kernel type	$\{0.1, 0.2, \cdots, 0.9, 1, 2, \cdots, 100\}$, {'linear', 'poly', 'rbf'}
MeanDecisiontree	min samples leaf	$\{1, 2, \cdots, 10\}$
MeanRandomForest	min samples leaf, the number of trees	$\{1, 2, \cdots, 10\}$, $\{5, 10, \cdots, 100\}$
MeanMLP	learning rate, hidden layer units, epochs	$\{0.0001, 0.001, 0.01, 0.1\}, \{20, 30, \cdots, 200\}, \{100, 200, 500, 1000, 1500\}$
DCCF [21]	bandwidth h_g	$\{1, 2, \cdots, 10, 20, \cdots, 50\}$
BCCF [21]	distance parameter δ	$\{0.1, 0.5, 1, 2, \cdots, 10\}$
DF-SVM	regularization parameter C, kernel type	$\{0.1, 0.2, \cdots, 0.9, 1, 2, \cdots, 100\}$, {'linear', 'poly', 'rbf'}
DF-MLP	learning rate, hidden layer units, epochs	$\{0.0001, 0.001, 0.01, 0.1\}, \{20, 30, \cdots, 200\}, \{100, 200, 500, 1000, 1500\}$



(a) (b) (c)

Fig. 2. Classification error rate on the test set varies with the number of synthetic data. (a) DF-SVM with four defuzzification functions. (b) DF-MLP with four defuzzification functions. (c) DF-SVM and DF-MLP with VAL.

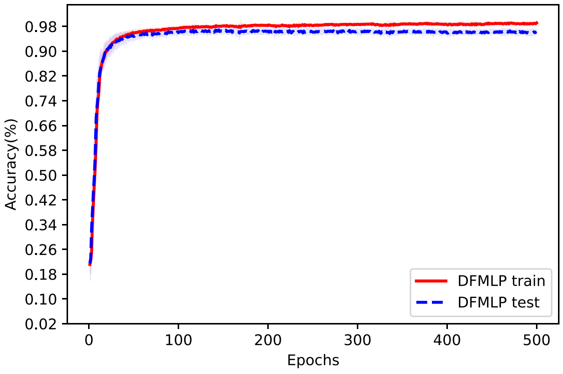


Fig. 3. Accuracy curve on the synthetic datasets versus the number of epochs.

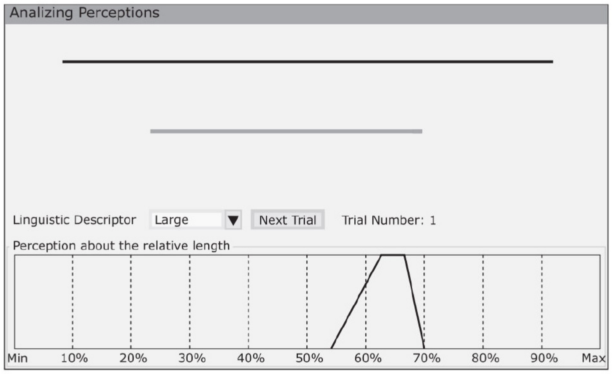
than the other three methods. The reason why VAL can achieve better performance than other methods is that VAL uses all information from fuzzy sets so that some key information is not discarded. In addition, VAL gives less importance to the lower levels of fuzzy sets, which is reasonable from the perspective of the concept of the membership function. Therefore, we use VAL as the defuzzification method in the following experiments. Moreover, from Fig. 2(c), it illustrates that the convergence rate of the two proposed algorithms with VAL defuzzification method is $O(1/\sqrt{m})$. Therefore, we confirmed the theoretical analysis results in Section V that we can always obtain a fuzzy classifier with high classification accuracy when sufficient fuzzy-feature observations are available.

The results of the second experiment are illustrated in Table II, and Fig. 3 shows the classification accuracy curve of Algorithm 2 on the synthetic datasets versus the number of epochs. From the results, DF-SVM and DF-MLP obtain better performance than the most other baselines on the synthetic dataset. Furthermore, the results of the statistic test show that DF-SVM outperforms other methods significantly at the 0.05 significance level, which demonstrates the superiority of the proposed algorithms. In addition, we present the experimental running times for the proposed algorithms and all baselines.

TABLE II
EXPERIMENT RESULT OF SYNTHETIC DATASET

Algorithms	Test accuracy	p	Time (sec)
Meanlogistic	96.86% ±0.87%	$2.2 \times 10^{-6*}$	119.97
MeanSVM	97.72% ±0.71%	0.0337*	127.35
MeanDecisiontree	78.20% ±2.70%	$6.3 \times 10^{-8*}$	2.23
MeanRandomForest	95.82% ±0.85%	$9.8 \times 10^{-8*}$	1088.57
MeanMLP	96.16% ±0.80%	$3.7 \times 10^{-7*}$	6607.89
DCCF [21]	92.58% ±1.02%	$6.3 \times 10^{-8*}$	1122687
BCCF [21]	92.51% ± 1.03%	$6.3 \times 10^{-8*}$	1123543
DF-SVM	**98.24% ± 0.52%**	—	119.98
DF-MLP	96.90% ± 0.95%	$2.2 \times 10^{-5*}$	6593.64

The bold value represents the highest accuracy in each column.
p: The p-value of the Wilcoxon rank-sum test between the performance of DF-SVM and other algorithms.
$^{*}p < 0.05$



Analizing Perceptions

Linguistic Descriptor: Large

Trial Number: 1

Perception about the relative length

Fig. 4. Software to evaluate the visual perception of a line segment.

VIII. Experiments on Real-World Datasets

In this section, five real-world datasets are used to verify the efficacy of proposed algorithms for solving the MCIMO problem by comparing with seven baselines in terms of classification accuracy. Besides, we show how to apply the proposed algorithms to analyze interval-valued datasets.

A. Real-World Datasets

In this section, we briefly introduce the five real-world datasets used in the experiments.

1) Perceptions Experiment Dataset: The 1st dataset, called the perceptions experiment dataset, contains 551 observations with one fuzzy feature. The fuzzy feature is a trapezoidal fuzzy number characterized by $(\inf P_0, \sup P_0, \inf P_1, \sup P_1)$. Each observation is the perceptions experiment result for one person. The description of perceptions experiment can be found in the following URL: http://bellman.ciencias.uniovi.es/SMIRE/Perceptions.html. In the perceptions experiment, the one black line that people will see is shown in Fig. 4. Once participants see a black line, they will be asked to give a trapezoidal fuzzy number characterized by $(\inf P_0, \sup P_0, \inf P_1, \sup P_1)$ to describe it.

For the first dataset, we consider using the fuzzy feature (i.e., the trapezoidal fuzzy number) to predict the category (very small; small; medium; large or very large), which will be selected by the participants according to their perception of the black line.

2) Mushroom Dataset: The 2nd dataset is the California mushroom dataset[1] that contains 245 instances in 17 fungi species categories. There are five interval-valued variables: the pileus cap width (X_1), the stipe length (X_2), the stipe thickness (X_3), the spores major axis length (X_4), and the spores minor axis length (X_5). Some instances of the mushroom dataset are shown in Table III. The goal of our experiment on this dataset is to predict the species category of the California mushroom using five interval-valued features.

TABLE III
Some Instances of the Mushroom Dataset

Species	X_1(cm)	X_2(cm)	X_3(cm)	X_4(cm)	X_5(μm)
Agaricus	[6,12]	[2,7]	[1.5,3]	[6,7.5]	[4,5]
Boletus	[7,14]	[5,9]	[3,6]	[11.5,13.5]	[3.5,4.5]
Amanita	[6,12]	[9,17]	[1,2]	[9.5,11.5]	[8.5,10]
Clitocybe	[2,9]	[2,6]	[0.5,1.2]	[5,6]	[2.5,3.5]

TABLE IV
Some Instances of the London Weather Data

Times	T	P0	P	U	Td	Y
31/12/2021	[0.8,6.1]	[730.2,733.4]	[755.5,759]	[76,99]	[0,3.3]	1
30/12/2021	[-1.4,1.5]	[734.2,735.8]	[759.8,762]	[77,93]	[-2.4,-0.6]	0
29/12/2021	[-1.2,2.1]	[730.5,735.4]	[756,761]	[93,97]	[-2.4,1.7]	1
28/12/2021	[-1.2,1.4]	[730.5,734.2]	[756.1,760]	[72,96]	[-4.2,0.1]	1

3) Letter Recognition Dataset: The 3rd dataset is the letter recognition dataset, selected from UCI machine learning repository (https://archive-beta.ics.uci.edu/), which contains 20 000 instances in 26 categories. This dataset contains 16 integer features extracted from raster scan images of the letters. We use the same methods described in Section VII to transfer integer features into fuzzy features. Then, we obtain one real-world dataset with fuzzy-valued features. The goal of our experiment on this dataset is to identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet.

4) London Weather Dataset: The 4th dataset is the meteorological data of London (from March 1, 2016 to December 31, 2021), provided by the "Reliable Prognosis" site (https://rp5.ru/), which contains 2131 instances. Each instance is meteorological data of one day in London, which described by five interval-valued variables (air temperature T, atmospheric pressure at weather station level $P0$, atmospheric pressure reduced to main sea level P, humidity U, and dew-point temperature Td) and one category variable (Precipitation or not: $0 \equiv$ No Precipitation, $1 \equiv$ Precipitation). Some instances of this dataset are shown in Table IV. We aim to use the five interval-valued features for precipitation prediction.

5) Washington Weather Dataset: The 5th dataset is the meteorological data of Washington (from January 1, 2016 to December 31, 2021) in the "Reliable Prognosis" site as well, which contains 2191 instances. Each instance is meteorological data of one day in Washington, which described by five interval-valued variables (same as the 4th dataset) and one category variable (same as the 4th dataset). We aim to use the five interval-valued features for precipitation prediction.

B. Preprocessing of Interval-Valued Data

We notice that the features of the 2nd, 4th, and 5th datasets are interval valued. Therefore, in this section, we present an approach to transform interval-valued features into fuzzy-valued features. Suppose $[A, B]$ is denoted as a feature of one interval-valued instance. Thus, we use one approach that maps $[A, B]$ to a triangular fuzzy number $\widetilde{x}$ characterized by $(A, \beta A + (1-\beta)B, B)$, where $\beta \in [0, 1]$ is a hyperparameter to control the shape of the membership function of $\widetilde{x}$.

[1] See https://www.mykoweb.com/CAF/ for more details.

TABLE V
EXPERIMENT RESULT OF PERCEPTIONS EXPERIMENT DATASET

Algorithms	Test accuracy	p
Meanlogistic	90.04% ±2.20%	0.0080*
MeanSVM	90.36% ±2.98%	0.5075
MeanDecisiontree	89.32% ±3.30%	0.0231*
MeanRandomForest	90.27% ±3.10%	0.3169
MeanMLP	90.45% ±2.91%	0.3793
DCCF [21]	87.82% ±2.15%	0.0001*
BCCF [21]	88.23% ± 2.01%	0.0001*
DF-SVM	91.00% ±2.52%	0.7251
DF-MLP	**91.50% ± 2.51%**	—

The bold value represents the highest accuracy in each column.
p: The p-value of the Wilcoxon rank-sum test between the performance of DF-MLP and other algorithms.
$^*p < 0.05$

TABLE VI
EXPERIMENT RESULT OF MUSHROOM DATASET

Algorithms	Balanced accuracy	AUC
Meanlogistic	71.36% ±3.86%	0.9645 ± 0.0079
MeanSVM	79.08% ±3.08%	0.9728 ± 0.0071
MeanDecisiontree	70.68% ±4.16%	0.9069 ± 0.0203
MeanRandomForest	79.04% ±3.83%	0.9750 ± 0.0077
MeanMLP	80.49% ±3.40%	0.9721 ± 0.0071
DCCF [21]	65.14% ±5.31%	0.9584 ± 0.0078
BCCF [21]	64.16% ±4.53%	0.9554 ± 0.0083
DF-SVM	81.71% ±4.44%	0.9758 ± 0.0103
DF-MLP	**83.57% ± 2.04%**	**0.9784 ± 0.0025**

The bold value represents the highest accuracy in each column.

TABLE VII
p-VALUE OF THE STATISTIC TEST ON MUSHROOM DATASET

Algorithms	Balanced accuracy	AUC
DF-MLP vs Meanlogistic	$6.3 \times 10^{-8*}$	0.0012*
DF-MLP vs MeanSVM	$3.5 \times 10^{-5*}$	0.4171
DF-MLP vs MeanDecisiontree	$6.3 \times 10^{-8*}$	$6.3 \times 10^{-8*}$
DF-MLP vs MeanRandomForest	0.0002*	0.0935
DF-MLP vs MeanMLP	0.0041*	0.6849
DF-MLP vs DCCF [21]	$6.3 \times 10^{-8*}$	$6.2 \times 10^{-5*}$
DF-MLP vs BCCF [21]	$6.3 \times 10^{-8*}$	$2.5 \times 10^{-5*}$
DF-MLP vs DF-SVM	0.1762	0.3438

$^*p < 0.05$

TABLE VIII
EXPERIMENT RESULT OF LETTER RECOGNITION DATASET

Algorithms	Test accuracy	p
Meanlogistic	73.50% ±0.70%	$6.3 \times 10^{-8*}$
MeanSVM	94.60% ±0.36%	0.0011*
MeanDecisiontree	78.09% ±0.69%	$6.3 \times 10^{-8*}$
MeanRandomForest	93.50% ±0.41%	$6.3 \times 10^{-8*}$
MeanMLP	91.79% ±0.47%	$6.3 \times 10^{-8*}$
DF-SVM	**95.01% ± 0.32%**	—
DF-MLP	93.61% ±0.43%	$6.3 \times 10^{-8*}$

The bold value represents the highest accuracy in each column.
p: The p-value of the Wilcoxon rank-sum test between the performance of DF-SVM and other algorithms.
$^*p < 0.05$

Through the above preprocessing, the DF-SVM and DF-MLP algorithms can be used to classify dataset with interval-valued instances. In addition, we realize that the second dataset is an imbalanced dataset, which means that each category contains a different number of instances. Therefore, a random oversampling technique (KMeansSMOTE [55]) is used to improve the performance of the proposed algorithms. After the process of the random oversampling technique, the data of each category in the second dataset is expanded to 30.

C. Experimental Setup

We use the same baselines in Section VII, and the experimental details of all methods are basically the same as in Section VII. The only difference is that one more hyperparameter β needs to be selected when analyzing the second dataset. We select the shape parameter β from $\{0, 0.05, 0.1, \ldots, 1\}$. Furthermore, we complete the Wilcoxon rank-sum tests of the method, which obtains the best performance, with other methods on real-world datasets. Since DCCF and BCCF cannot well handle the dataset with a large number of instances, we only compare the proposed algorithms with the first five baselines on the last three datasets in our experiments.

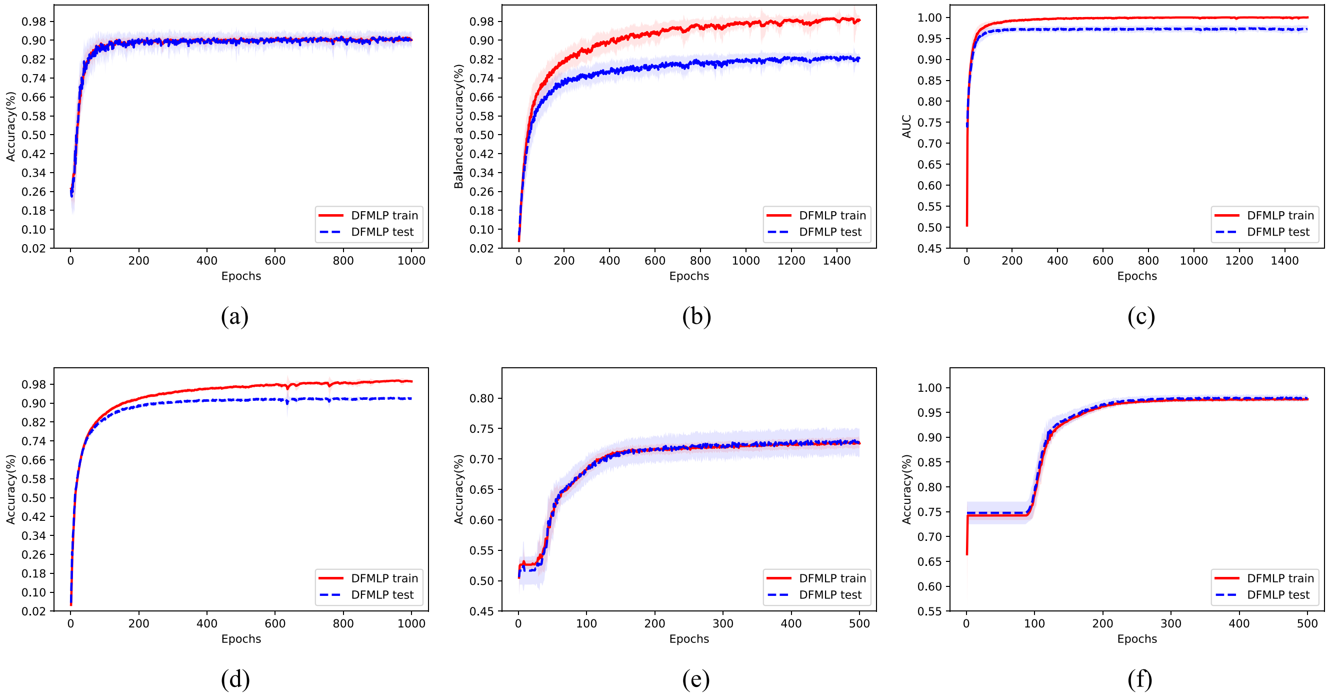
In addition, since the second dataset is an imbalanced dataset, we use balanced accuracy [56] and *AUC* instead of classification accuracy to compare model performance on the second dataset. The definition of balanced accuracy is

$$\text{Balanced Accuracy} = \frac{1}{K}\sum_{k=1}^{K}(\text{Recall of } k\text{−th class})$$
$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. *AUC* is equal to the compute area under the receiver operating characteristic curve.

D. Experimental Results Analysis

All the experiment results on the five real-world datasets are illustrated in Tables V–X, and how the evaluation metrics varies with the number of epochs for Algorithm 2 are shown in Fig. 5. From these results, the proposed two algorithms achieve better performance than other baselines on all five real-world datasets, which illustrates the efficacy of the proposed algorithms in addressing real-world datasets with fuzzy-valued or interval-valued features. Moreover, the results of the statistic test show that the proposed two algorithms outperform most other methods significantly at the 0.05 significance level, which demonstrates the superiority of the proposed algorithms. Furthermore, for the 1st, 2nd, and 5th datasets, DF-MLP obtains the highest average performance on the test set. While, for the letter recognition dataset and London weather dataset,



(a) (b) (c)

(d) (e) (f)

Fig. 5. Evaluation metrics varies with the number of epochs. (a) DF-MLP on the perceptions experiment dataset. (b) DF-MLP on the mushroom dataset. (c) DF-MLP on the mushroom dataset. (d) DF-MLP on the letter recognition dataset. (e) DF-MLP on the London weather dataset. (f) DF-MLP on the Washington weather dataset.

TABLE IX
EXPERIMENT RESULT OF LONDON WEATHER DATASET

Algorithms	Test accuracy	p
Meanlogistic	71.58% ±1.94%	0.0038*
MeanSVM	72.26% ±2.15%	0.049*
MeanDecisiontree	69.11% ±1.99%	$1.5 \times 10^{-5*}$
MeanRandomForest	72.76% ±1.84%	0.24
MeanMLP	71.53% ±2.10%	0.00059*
DF-SVM	$\mathbf{73.55\% \pm 1.73\%}$	—
DF-MLP	73.06% ±1.91%	0.33

The bold value represents the highest accuracy in each column.
p: The p-value of the Wilcoxon rank-sum test between the performance of DF-SVM and other algorithms.
$^*p < 0.05$

TABLE X
EXPERIMENT RESULT OF WASHINGTON WEATHER DATASET

Algorithms	Test accuracy	p
Meanlogistic	97.60% ±0.60%	0.045*
MeanSVM	97.76% ±0.66%	0.30
MeanDecisiontree	97.26% ±0.74%	0.0026*
MeanRandomForest	97.34% ±0.74%	0.0043*
MeanMLP	97.65% ±0.52%	0.049*
DF-SVM	97.95% ±0.66%	0.90
DF-MLP	$\mathbf{98.01\% \pm 0.62\%}$	—

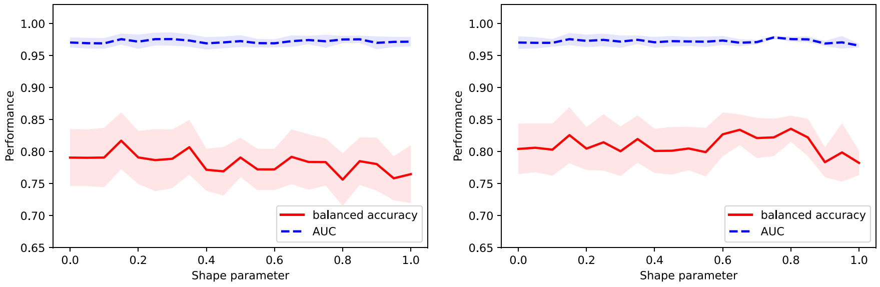
The bold value represents the highest accuracy in each column.
p: The p-value of the Wilcoxon rank-sum test between the performance of DF-SVM and other algorithms.
$^*p < 0.05$

DF-SVM is more prioritized than other methods, which means that the proposed algorithms are applicable to different types of datasets.

E. Parameters Sensitivity Analysis

In this section, we analyze whether the value of the shape parameter β in DF-SVM and DF-MLP affects the balanced accuracy and *AUC* on the mushroom dataset.

We conduct the same preprocessing for the mushroom dataset. We select the shape parameter β from $\{0, 0.05, 0.1, \ldots, 1\}$. Then, for each value of β, the results are obtained using the same experimental operation in Section VII. Fig. 6(a) and (b) shows the mean and standard deviation of the balanced accuracy and *AUC* of the test sets on the mushroom dataset when the shape parameter β of both algorithms changes from 0 to 1. These figures illustrate that a different value for the shape parameter β will affect the classification performance since the value of β determines the shape of the triangular fuzzy number. A value of β that can achieve high performance means that the proposed algorithms with this value of β can extract more significant information from the datasets with fuzzy-valued or interval-valued features. Therefore, we can improve the performance of DF-SVM and DF-MLP by finding a suitable value of β. In our experiments, we find the optimal value of β in the validation set.



(a)

(b)

Fig. 6. Evaluation metrics of the test sets varies with the value of shape parameter β. (a) DF-SVM. (b) DF-MLP.

IX. Conclusion and Future Work

In this article, we identified a new problem called MCIMO. In the MCIMO problem, we need to train a fuzzy classifier when only fuzzy-feature observations are available.

First, we identified a novel problem called MCIMO in Section IV. Since there are no existing papers for theoretical analysis of fuzzy classifiers, we give the estimation error bounds for the MCIMO problem in this article. These bounds illustrate that we can always train a fuzzy classifier with high classification accuracy to solve the MCIMO problem as long as sufficient fuzzy-feature instances can be collected.

Hence, two algorithms are constructed to handle the MCIMO problem. In addition, the optimal defuzzification function for the proposed fuzzy technique-based algorithms is found by comparing the performance of different defuzzification methods on synthetic datasets. Finally, experimental results on synthetic datasets and three real-world datasets show the superiority of the proposed algorithms. Moreover, through comparisons with several nonfuzzy baselines, the experimental results demonstrate that the proposed fuzzy-based methods can obtain better performance in analyzing fuzzy data or interval-valued data than nonfuzzy methods. Since they use fuzzy vectors to express the distribution of imprecise data and apply different defuzzification methods to extract crisp-valued information from imprecise observations.

In future research, we plan to study more complicated issues, for example, covariate shift and domain adaptation with imprecise observations. We can obtain the theoretical analysis and solutions of these issues based on the introduced theoretical analysis and algorithms in this article. In addition, we found that the proposed two algorithms can obtain better performance in processing interval-valued data. Therefore, we consider analyzing interval-valued data based on the proposed two algorithms in future studies.

References

[1] P. Seeböck *et al.*, “Unsupervised identification of disease marker candidates in retinal OCT imaging data,” *IEEE Trans. Med. Imag.*, vol. 38, no. 4, pp. 1037–1047, Apr. 2019.

[2] R. Xia, C. Zong, X. Hu, and E. Cambria, “Feature ensemble plus sample selection: Domain adaptation for sentiment classification,” *IEEE Intell. Syst.*, vol. 28, no. 3, pp. 10–18, May/Jun. 2013.

[3] X. Zhu, H. Suk, S. Lee, and D. Shen, “Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification,” *IEEE Trans. Biomed. Eng.*, vol. 63, no. 3, pp. 607–618, Mar. 2016.

[4] L. Nanni, S. Ghidoni, and S. Brahnam, “Handcrafted vs. non-handcrafted features for computer vision classification,” *Pattern Recognit.*, vol. 71, pp. 158–172, Nov. 2017.

[5] G. Wang, J. Lu, K.-S. Choi, and G. Zhang, “A transfer-based additive LS-SVM classifier for handling missing data,” *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 739–752, Feb. 2020.

[6] C. P. Chen, Y.-J. Liu, and G.-X. Wen, “Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems,” *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 583–593, May 2014.

[7] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: Massachusetts Inst. Technol., 2012.

[8] V. Koltchinskii and D. Panchenko, “Empirical margin distributions and bounding the generalization error of combined classifiers,” *Ann. Stat.*, vol. 30, no. 1, pp. 1–50, 2002.

[9] Y. Maximov, M.-R. Amini, and Z. Harchaoui, “Rademacher complexity bounds for a Penalized multi-class semi-supervised algorithm,” *J. Artif. Intell. Res.*, vol. 61, no. 1, pp. 761–786, Jan. 2018.

[10] E. L. Allwein, R. E. Schapire, and Y. Singer, “Reducing Multiclass to binary: A unifying approach for margin classifiers,” *J. Mach. Learn. Res.*, vol. 1, no. 2, pp. 113–141, 2000.

[11] A. Daniely and S. Shalev-Shwartz, “Optimal learners for multiclass problems,” in *Proc. Conf. Learn. Theory*, 2014, pp. 287–316.

[12] M. Hardt, B. Recht, and Y. Singer, “Train faster, generalize better: Stability of stochastic gradient descent,” in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1225–1234.

[13] D. McAllester, “A PAC-Bayesian tutorial with a dropout bound,” 2013, *arXiv:1307.2118*.

[14] C. Xu, T. Liu, D. Tao, and C. Xu, “Local rademacher complexity for multi-label learning,” *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1495–1507, Mar. 2016.

[15] J. Li, Y. Liu, R. Yin, H. Zhang, L. Ding, and W. Wang, “Multi-class learning: From theory to algorithm,” in *Proc. NeurIPS*, vol. 31, 2018, pp. 1593–1602.

[16] Y. Zhang, F. Liu, Z. Fang, B. Yuan, G. Zhang, and J. Lu, “Clarinet: A one-step approach towards budget-friendly unsupervised domain adaptation,” 2020, *arXiv:2007.14612*.

[17] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang, “Open set domain adaptation: Theoretical bound and algorithm,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4309–4322, Oct. 2021.

[18] L. Zhong, Z. Fang, F. Liu, J. Lu, B. Yuan, and G. Zhang, “How does the combined risk affect the performance of unsupervised domain adaptation approaches?” in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 11079–11087.

[19] Z. Fang, J. Lu, F. Liu, and G. Zhang, “Semi-supervised heterogeneous domain adaptation: Theory and algorithms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 27, 2022, doi: 10.1109/TPAMI.2022.3146234.

[20] J. Dong, Y. Cong, G. Sun, Z. Fang, and Z. Ding, “Where and how to transfer: Knowledge aggregation-induced transferability perception for unsupervised domain adaptation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 16, 2021, doi: 10.1109/TPAMI.2021.3128560.

[21] A. Colubi, G. González-Rodríguez, M. Ángeles Gil, and W. Trutschnig, “Nonparametric criteria for supervised classification of fuzzy data,” *Int. J. Approx. Reason.*, vol. 52, no. 9, pp. 1272–1282, 2011.

[22] G. Wang, T. Zhou, K.-S. Choi, and J. Lu, “A deep-ensemble-level-based interpretable Takagi–Sugeno–Kang fuzzy classifier for imbalanced data,” *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3805–3818, May 2022.

[23] F. Liu, G. Zhang, and J. Lu, "Multi-source heterogeneous unsupervised domain adaptation via fuzzy-relation neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 11, pp. 3308–3322, Nov. 2021.

[24] H. Zuo, J. Lu, G. Zhang, and F. Liu, "Fuzzy transfer learning using an infinite Gaussian mixture model and active learning," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 2, pp. 291–303, Feb. 2019.

[25] J. Lu, H. Zuo, and G. Zhang, "Fuzzy multiple-source transfer learning," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 12, pp. 3418–3431, Dec. 2020.

[26] G. Ma, F. Liu, G. Zhang, and J. Lu, "Learning from imprecise observations: An estimation error bound based on fuzzy random variables," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2021, pp. 1–8.

[27] M. L. Puri and D. A. Ralescu, "Fuzzy random variables," *J. Math. Anal. Appl.*, vol. 114, no. 2, pp. 409–422, 1986.

[28] H. C. Wu, "Probability density functions of fuzzy random variables," *Fuzzy Sets Syst.*, vol. 105, no. 1, pp. 139–158, 1999.

[29] B. Sinova, M. Á. Gil, M. T. López, and S. V. Aelst, "A parameterized L_2 metric between fuzzy numbers and its parameter interpretation," *Fuzzy Sets Syst.*, vol. 245, pp. 101–115, Jun. 2014.

[30] X. Yang, G. Zhang, J. Lu, and J. Ma, "A kernel fuzzy C-means clustering-based fuzzy support vector machine algorithm for classification problems with outliers or noises," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 105–115, Feb. 2011.

[31] Y. Rong, Z. Wang, P. A. Heng, and K. S. Leung, "Classification of heterogeneous fuzzy data by Choquet integral with fuzzy-valued integrand," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 5, pp. 931–942, Oct. 2007.

[32] F. Liu, G. Zhang, and J. Lu, "A novel non-parametric two-sample test on imprecise observations," in *Proc. FUZZ-IEEE*, 2020, pp. 1–6.

[33] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, 2012.

[34] F. Liu, W. Xu, J. Lu, G. Zhang, A. Gretton, and D. J. Sutherland, "Learning deep kernels for non-parametric two-sample tests," in *Proc. ICML*, 2020, pp. 6316–6326.

[35] F. Liu, W. Xu, J. Lu, and D. J. Sutherland, "Meta two-sample testing: Learning kernels for testing with limited data," in *Proc. NeurIPS*, 2021, pp. 5848–5860.

[36] V. Behbood, J. Lu, and G. Zhang, "Fuzzy refinement domain adaptation for long term prediction in banking ecosystem," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1637–1646, May 2014.

[37] V. Behbood, J. Lu, G. Zhang, and W. Pedrycz, "Multistep fuzzy bridged refinement domain adaptation algorithm and its application to bank failure prediction," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 1917–1935, Dec. 2015.

[38] C. Yang, Z. Deng, K.-S. Choi, and S. Wang, "Takagi–Sugeno–Kang transfer learning fuzzy logic system for the adaptive recognition of epileptic electroencephalogram signals," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 5, pp. 1079–1094, Oct. 2016.

[39] Z. Deng, P. Xu, L. Xie, K.-S. Choi, and S. Wang, "Transductive joint-knowledge-transfer TSK FS for recognition of epileptic EEG signals," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 8, pp. 1481–1494, Aug. 2018.

[40] L. Xie, Z. Deng, P. Xu, K.-S. Choi, and S. Wang, "Generalized hidden-mapping transductive transfer learning for recognition of epileptic electroencephalogram signals," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2200–2214, Jun. 2018.

[41] Y. Jiang, Y. Zhang, C. Lin, D. Wu, and C.-T. Lin, "EEG-based driver drowsiness estimation using an online multi-view and transfer TSK fuzzy system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1752–1764, Mar. 2021.

[42] F. Liu, J. Lu, and G. Zhang, "Unsupervised heterogeneous domain adaptation via shared fuzzy equivalence relations," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3555–3568, Dec. 2018.

[43] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, and S. Wang, "Knowledge-leverage-based fuzzy system and its modeling," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 4, pp. 597–609, Aug. 2013.

[44] Z. Deng, Y. Jiang, K.-S. Choi, F.-L. Chung, and S. Wang, "Knowledge-leverage-based TSK fuzzy system modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1200–1212, Aug. 2013.

[45] Z. Deng, Y. Jiang, H. Ishibuchi, K.-S. Choi, and S. Wang, "Enhanced knowledge-leverage-based TSK fuzzy system modeling for inductive transfer learning," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 1, pp. 1–21, 2016.

[46] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Granular fuzzy regression domain adaptation in Takagi–Sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 847–858, Apr. 2018.

[47] B. Quost and T. Denœux, "Clustering and classification of fuzzy data using the fuzzy EM algorithm," *Fuzzy Sets Syst.*, vol. 286, pp. 134–156, Mar. 2016.

[48] P. D'Urso and J. M. Leski, "Fuzzy clustering of fuzzy data based on robust loss functions and ordered weighted averaging," *Fuzzy Sets Syst.*, vol. 389, pp. 1–28, Jun. 2020.

[49] S. Roychowdhury and W. Pedrycz, "A survey of defuzzification strategies," *Int. J. Intell. Syst.*, vol. 16, no. 6, pp. 679–695, 2001.

[50] W. Van Leekwijck and E. E. Kerre, "Defuzzification: Criteria and classification," *Fuzzy Sets Syst.*, vol. 108, no. 2, pp. 159–178, 1999.

[51] M. Oussalah, "On the compatibility between defuzzification and fuzzy arithmetic operations," *Fuzzy Sets Syst.*, vol. 128, no. 2, pp. 247–260, 2002.

[52] M. Delgado, M. A. Vila, and W. Voxman, "On a canonical representation of fuzzy numbers," *Fuzzy Sets Syst.*, vol. 93, no. 1, pp. 125–135, 1998.

[53] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proc. Eur. Symp. Artif. Neural Netw.*, vol. 99, 1999, pp. 219–224.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–9.

[55] F. Last, G. Douzas, and F. Bacao, "Oversampling for imbalanced learning based on *k*-means and smote," 2017, *arXiv:1711.00837*.

[56] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proc. IEEE 20th Int. Conf. Pattern Recognit.*, 2010, pp. 3121–3124.

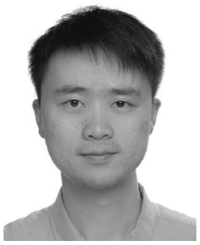
Guangzhi Ma received the B.S. degree in mathematics and applied mathematics from the School of Mathematics Sciences, Anhui University, Anhui, China, in 2017, and the M.S. degree in probability and statistics from the School of Mathematics and Statistics, Lanzhou University, Lanzhou, China, in 2020. He is currently pursuing the Ph.D. degree with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia.

He is a member of the Decision Systems and e-Service Intelligence Laboratory, Australia Artificial Intelligence Institute, University of Technology Sydney. His research interests include fuzzy transfer learning and domain adaptation.

Jie Lu (Fellow, IEEE) received the Ph.D. degree from Curtin University, Perth, WA, Australia, in 2000.

She is a Distinguished Professor and the Director of Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW, Australia. She has published over 500 papers in IEEE Transactions and other journals and conferences, and supervised 50 Ph.D. students to completion. Her main research expertise is in transfer learning, concept drift, decision support systems, and recommender systems.

Dr. Lu has received the UTS Medal for research excellence in 2019, the IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award in 2019, and the Australian Most Innovative Engineer Award in 2019. She has been awarded over ten Australian Research Council discovery grants and led 20 industry projects. She serves as an Editor-in-Chief for *Knowledge-Based Systems* (Elsevier) and *International Journal on Computational Intelligence Systems* (Springer). She has delivered 35 keynote speeches at international conferences. She is an IFSA Fellow and the Australian Laureate Fellow.



Feng Liu (Member, IEEE) received the B.Sc. degree in mathematics and the M.Sc. degree in probability and statistics from the School of Mathematics and Statistics, Lanzhou University, Lanzhou, China, in 2013 and 2015, respectively, and the Ph.D. degree in computer science from the University of Technology Sydney, Sydney, NSW, Australia, in 2020.

He is a Lecturer with the Australian Artificial Intelligence Institute, Faculty of Engineering and Information Technology, University of Technology Sydney. His research interests include hypothesis testing and trustworthy machine learning.

Dr. Liu has received the Outstanding Reviewer Awards of ICLR in 2021 and NeurIPS in 2021, the UTS-FEIT HDR Research Excellence Award in 2019, and the Best Student Paper Award of FUZZ-IEEE in 2019. He has served as a Senior Program Committee Member for ECAI and a Program Committee Member for NeurIPS, ICML, AISTATS, ICLR, KDD, AAAI, IJCAI, and FUZZ-IEEE. He also served as a reviewer for *Journal of Machine Learning Research*, *Machine Learning*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Zhen Fang (Member, IEEE) received the M.Sc. degree in pure mathematics from the School of Mathematical Sciences, Xiamen University, Xiamen, China, in 2017. He is currently pursuing the Ph.D. degree with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia.

He is a member of the Decision Systems and e-Service Intelligence Research Laboratory, Australian Artificial Intelligence Institute, University of Technology Sydney. He has published several paper related to transfer learning and out-of-distribution learning in IJCNN, NeurIPS, AAAI, IJCAI, ICML, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. His research interests include transfer learning and out-of-distribution learning.

Guangquan Zhang received the Ph.D degree in applied mathematics from Curtin University, Perth, WA, Australia, in 2001.

He is an Australian Research Council QEII Fellow, and an Associate Professor and the Director of the Decision Systems and e-Service Intelligent Research Laboratory, Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, NSW, Australia. From 1993 to 1997, he was a Full Professor with the Department of Mathematics, Hebei University, Baoding, China. He has published six authored monographs, five edited research books, and over 500 papers including some 300 refereed journal articles. His main research interests lie in the area of fuzzy multiobjective, bilevel and group decision making, fuzzy measure, and machine learning.

Dr. Zhang has won ten ARC Discovery Project grants and many other research grants, and supervised 35 Ph.D. students to completion. He has served as a guest editor for special issues of IEEE Transactions and other international journals.