Fuzzy Ordered *c*-Means Clustering and Least Angle Regression for Fuzzy Rule-Based Classifier: Study for Imbalanced Data

Jacek M. Leski, *Senior Member, IEEE*, Robert Czabański, Michal Jezewski, and Janusz Jezewski, *Senior Member, IEEE*

***Abstract*—This article introduces a new classifier design method that is based on a modification of the traditional fuzzy clustering. First, a new fuzzy ordered *c*-means clustering is proposed. This method can be considered as a generalization of the concept of the conditional fuzzy clustering by introducing ordering and weighting distances from data to cluster prototypes. As a result, a more local impact of data on created groups and increased repulsive force between group prototypes are obtained. The proposed method provides a better representation of the data classes, in particular for classes with small cardinality in the training set (imbalanced data). A special initialization of the prototypes is also introduced. Next, the proposed clustering method is used to construct the premises of if–then rules of a fuzzy classifier. The conclusions of the rules are obtained by the least angle regression algorithm, which selects only those rules, that maximize the generalization ability of a classifier. Each if–then rule is represented in easily interpretable Mamdani–Assilian form. Finally, an extensive experimental analysis on 89 benchmark balanced and imbalanced datasets is performed to demonstrate the validity of the introduced classifier. Its competitiveness to state-of-the-art classifiers, with respect to both performance and interpretability, is shown as well.**

***Index Terms*—Conditional fuzzy clustering, fuzzy classifier design, imbalanced data, rulebase with good interpretability.**

I. Introduction

PATTERN recognition deals with the classification of objects (more precisely, patterns which describe these objects) into a certain number of predefined categories (classes). This field of study has recently played an important role in many engineering fields, such as data mining, medical diagnosis, character recognition, communication, computer vision, and many others [1]–[5]. There are two big challenges when a classifier is designed from real-world data: interpretability-generalization ability tradeoff and dealing with class imbalance. The most important feature of a classifier is its generalization ability which consists in producing reasonable decisions for data that have not been used in the process of the classifier design (so-called training or learning) [6]. The usual way to measure the generalization ability is cross-validation, i.e., measurement of the quality of the classification using a test set containing data that do not belong to the training set. Reiteration of the process of learning/testing, called multifold cross-validation, is usually used [6], [7]. Another important feature of a classifier is its interpretability which can be understood as the capability to express the classifier knowledge base in a manner that would be understandable for humans (domain experts) [8].

Class imbalance often occurs in real-world data. It consists in dominating the number of examples in the learning/testing set by one of the class, called majority class (skewed distribution of data). The degree of class disproportion is described by imbalance ratio (IR) defined as the cardinality of the majority class divided by the cardinality of the minority class. For example, in the medical diagnosis, the most common are healthy people, and the correct detection of persons at risk of disease is of maximum importance. Similarly, we need an efficient method to detect spam in messages received via email. Hence, the growing interest in methods of classifier design, that would be resistant to imbalanced classes, is observed. Generally, existing methods can be divided into [9]: operating on data (data-level, external) and adapting learning algorithms (algorithm-level, internal).

Methods operating on the data rely on artificial equalizing cardinality of classes (balancing data). There are methods of down-sampling the majority classes and/or oversampling the minority classes [10]. Both approaches can be easily criticized, the first one can remove important information from the training set, while the second can lead to overfitting by duplicating examples. Methods which adapt learning algorithms rely on increasing attention to examples from minority classes, by using selected weights [11]–[14]. The basic difficulty is the appropriate selection of the weights. This article proposes a method to adopt the learning algorithm, but without using weights. Resistance to imbalanced classes is included in the method structure itself.

The measurement of correct classifications (overall accuracy) is usually used to assess the generalization ability, however

Manuscript received January 26, 2019; revised May 3, 2019 and August 8, 2019; accepted August 20, 2019. Date of publication October 9, 2019; date of current version October 30, 2020. This work was supported in part by the National Science Centre in Poland under Grant 2017/27/B/ST6/01989. The works of J. M. Leski, R. Czabanski, and M. Jezewski were supported in part by statutory funds (BK-2018, BK-2019) of the Institute of Electronics, Silesian University of Technology. *(Corresponding author: Jacek M. Leski.)*

J. M. Leski is with the Silesian University of Technology, 44-100 Gliwice, Poland, and also with the Łukasiewicz Research Network–Institute of Medical Technology and Equipment, 41-800 Zabrze, Poland (e-mail: jleski@polsl.pl).

R. Czabański and M. Jezewski are with the Silesian University of Technology, 44-100 Gliwice, Poland (e-mail: robert.czabanski@polsl.pl; michal.jezewski@polsl.pl).

J. Jezewski is with the Łukasiewicz Research Network–Institute of Medical Technology and Equipment, 41-800 Zabrze, Poland (e-mail: jezewski@itam.zabrze.pl).

Digital Object Identifier 10.1109/TFUZZ.2019.2939989

it is not suitable for imbalanced data. If for example, a class representing the group of healthy people constitutes 95% of all available data, then the criterion based on the correct classifications allows a trivial classifier, which by assigning all people to a healthy class achieves a very good quality of classification equal to 95%! Hence, usually a G-mean index is used to evaluate the quality of classification for imbalanced data. The G-mean is defined as the geometric mean of the positive accuracy (sensitivity) and the negative accuracy (specificity).

Until now, a lot of classifier design methods have been proposed. An overview of these methods can be found in classical monographs: [2], [4], and [5]. The support vector machine (SVM) [7], the AdaBoost [15], and the kernel Fisher discriminant (KFD) [16] are among the most successful ones. The main disadvantages of these state-of-the-art techniques are their high computational cost and poor interpretability. To overcome the computational cost drawback, a number of alternative methods have been proposed, including: the Lagrangian support vector machine (LSVM) [17], the least squares SVM [18], and the core vector machine [19]–[21]. To overcome the poor interpretability drawback, fuzzy rule-based classifiers are usually used, but their performance (generalization ability) is often worse when comparing with best classifiers. Therefore, the goal of this work is to introduce a classifier design method with a human-readable knowledge base (high interpretability), while maintaining high generalization ability. The method should also have a high overall accuracy for balanced datasets and a high G-mean value for imbalanced ones.

A fuzzy classifier allows for finding a nonlinear decision function of inputs (features of the recognized objects) with the knowledge base expressed by a set of the conditional sentences with linguistic interpretability, i.e., if–then rules with linguistic premises and conclusions [22]–[27]. Similarly to fuzzy systems, fuzzy classifiers can be divided into: 1) the Mamdani–Assilian classifiers (MAC) in which the linguistic terms are used in both premises and conclusions, and 2) the Takagi–Sugeno–Kang classifiers (TSKC) in which premises have linguistic terms, however conclusions are functions of input features. It is well known from the literature that the TSKC have the greater generalization ability, but their interpretability for humans is poor [8], [24], [27], [28]. In contrast, MAC are easily interpretable, but their generalization ability is weak. However, [29] shows that it is possible to obtain a good balance between the contradictory features of a fuzzy system: its generalization ability and interpretability. The method is based on the statement well known from the approximation theory [30] and the machine learning [7]—too accurate learning on a training set leads to overfitting (also known as overtraining) which results in a poor generalization ability. In other words, we should limit learning accuracy to obtain competitive generalization ability. Limiting accuracy is an intrinsic characteristic of fuzzy systems, which causes a natural opportunity to increase their generalization ability. This article proposes the two-class (binary) classifier, however, the method can easily be generalized to a multiclass problem using the "one-against-one" (all-versus-all, class-class) or the "one-against-all" (one-versus-rest, class-remainder) methodologies [3], [31].

To obtain the premises of the if–then rules for both TSKC and MAC, the fuzzy clustering of the training data in the input space is most commonly applied [24], [26], [32], [33]. To obtain premises for one class, the data from the training set that belong to this class are selected and undergo clustering. The data from other classes are not taken into consideration. In [29], another approach called fuzzy $(C + P)$-means (FCPM) clustering was presented, where the data structure from one class was searched taking into account the structure of other classes. This method was used to find premises of if–then rules followed by the iteratively reweighted least squares (IRLS) procedure to obtain conclusions of the rules with the certainty factors. The FCPM method is based on the idea that the data from other classes should have an influence on the prototypes of the clustered class. These prototypes should be attracted to the regions where the data from this class are dense and simultaneously repulsed from the data from other classes. This approach works well for balanced data. However, for the imbalanced data, the class with a higher cardinality (majority class) always plays the dominant role in the competition between prototypes for the feature space. Modification of the above method for imbalanced data was presented in [34], where the differentiation of influence of various classes was aimed to increase the chance of classes with a low cardinality (minority classes) in determining the decision function. Another approach has been used in this article. To determine the premises of the rules, data from the minority and the majority classes are grouped into the same number of groups using a specially designed clustering method called fuzzy ordered C-means clustering (FOCM). The basic feature of this method is to increase the repulsive force between group prototypes, through a more local impact of data on created groups (ordering and weighting distances from cluster prototypes). In this way, we get a better representation of the data class, in particular for classes with small cardinality in the training set (imbalanced data). As a result, an excessive number of rules arise, which are reduced in the second step—adjusting the rule conclusions (their certainty factors). At this stage the least angle regression (LAR) algorithm is used to select only those if–then rules, that maximize the generalization ability.

Summarizing, our main motivation is the proposal of a new classifier design method, resulting from the synergy of the FOCM clustering and the LAR algorithm, for achieving high generalization ability when evaluating the imbalance data, while maintaining the possibility of interpreting the learning outcomes thanks to the linguistic representation of the knowledge in the form of fuzzy conditional (if–then) rules. Consequently, the goal of this article is multifold: 1) it introduces a new fuzzy clustering called fuzzy ordered c-means; 2) it uses this clustering and the LAR algorithm in a new method of rule-based classifier design; 3) it investigates the generalization ability of the designed fuzzy classifiers for real-world high-dimensional benchmark data (balanced and imbalanced); and 4) it compares their generalization ability (with a good interpretability) to state-of-the-art classifiers.

The remainder of this article is organized as follows: Section II shows fuzzy clustering with the impact of data on the location of prototypes, which depends on the order of distances between the data and the prototypes. A special initialization of the clustering (determination of the initial prototypes) is also outlined in this section. Section III describes an application of the abovementioned clustering method for designing a fuzzy classifier, in particular to determine the premises of its if–then rules. Section IV presents the use of the LAR algorithm to the design rule conclusions of a binary classifier. In Section V, the experiments are presented and the results of the real-world and the synthetic benchmark datasets classification are discussed. Finally, Section VI concludes this article.

II. New Conditional Clustering Method—Fuzzy Ordered c-Means Clustering

Fuzzy clustering methods consist in soft dividing a set of objects into clusters in such a way that the members of the same cluster are more similar to each other than to the members of the other clusters. The idea has been introduced by Ruspini and used by Dunn to construct a fuzzy clustering method based on the criterion function minimization (see [35] and its references list). One of the most popular clustering methods based on this approach is the fuzzy C-means (FCM) introduced by Bezdek [36]. This powerful algorithm has been successfully applied to a wide variety of problems. In the literature, there are many modifications of the FCM method. In this section, a modification useful for a rule-based classifier design is introduced.

In the traditional FCM method prototypes are close to as many data points as possible, and the so-called probabilistic constraint ensures a repulsive force between prototypes, leading to prototypes variety. Therefore clustering is considered as a method searching for the data structure. The analyzed data are described using a small number of prototypes. A very important application of fuzzy clustering is extracting the fuzzy if–then rules. Most often the clustering is used only to find the premises of the rules. For classifier design, this search is performed on a training set that contains examples from all classes. Of course, if we consider imbalanced data, then clustering of all classes leads to the dominance of a class with a greater cardinality. In other words, a larger number of if–then rules will describe the majority class. Therefore, for imbalanced data the clustering should be performed separately for each class. Consequently, the training data from one class only are selected for clustering, and the remaining data from other classes are not taken into account. The time of necessary computations increases due to the iterative nature of the clustering algorithm, but it is somewhat balanced by a smaller size of the clustered dataset. Such an approach causes that even small classes are represented by a predetermined number of rules. To ensure the greatest possible variety of prototypes, the force that repels prototypes from each other should be as large as possible. In the FCM method this force comes from a probabilistic constraint and it can be additionally increased by decreasing the value of the weighting exponent toward value 1. In the work [29] a value of 1.1 was used, which increased the repulsive force between prototypes. However, greater reduction in weighting exponent value leads to instability of the method and additional solution should be used. The modification of the FCM method proposed in this article is based on the idea that a datum should only affect the closest groups (the corresponding rules). In this way, the groups will become less fuzzy, what can be interpreted as they are repulsed with greater strength. In the proposed approach the distances of a datum to all prototypes will be ordered and the associated weights will depend on this order, i.e., prototypes with smaller distances will have higher weight values. Another approach to clustering using ordered data was applied in [37], where in the fuzzy C-ordered means (FCOM) method data are ordered with respect to the distance from given group's prototype.

If we have c prototypes, then the proposed method can be called as the Fuzzy Ordered C-Means (FOCM) clustering. Clustering methods divide a set of N observations (input vectors) $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^n$ into c groups denoted as $\Omega_1, \Omega_2, \ldots, \Omega_c$. The criterion function of the proposed fuzzy ordered c-means can be written as

$$J(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{c} \sum_{k=1}^{N} \beta_{i,k} \left(u_{\pi_k(i),k}\right)^m \, d^2_{\pi_k(i),k} \tag{1}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c] \in \mathbb{R}^{n \times c}$ is a matrix consisting of the constructed prototypes and $\mathbf{U} = [u_{i,k}] \in [0,1]^{c \times N}$ is the partition matrix. The $u_{i,k}$ stands for the membership degree of the vector $\mathbf{x}_k$ to the cluster represented by the prototype $\mathbf{v}_i$. The squared Euclidean norm is denoted as $d^2_{i,k} = \|\mathbf{x}_k - \mathbf{v}_i\|^2$. Let $\pi_k(i) : \{1, 2, \ldots, c\} \to \{1, 2, \ldots, c\}$ be the permutation function. For each k the rank-ordered distances satisfy the following conditions:

$$d^2_{\pi_k(1),k} \leq d^2_{\pi_k(2),k} \leq d^2_{\pi_k(3),k} \leq \cdots \leq d^2_{\pi_k(c),k}. \tag{2}$$

If for each k, $\beta_{i,k}$ parameters fulfill $\beta_{1,k} \geq \beta_{2,k} \geq \cdots \geq \beta_{c,k}$, then it is clear that the impact of $\mathbf{x}_k$ to distant groups is reduced by down-weighting the respective products $(u_{\pi_k(i),k})^m \, d^2_{\pi_k(i),k}$. Generally, parameters $\beta_{i,k}$ may depend on k, i.e., may be selected for each $\mathbf{x}_k$. However, for simplicity, we assume in the further part of the work that $\beta_{i,k} = \beta_i$. The form of parameters β_i is proposed to be exponential

$$\beta_i = \exp\left(-\gamma \frac{i-1}{c-1}\right) \tag{3}$$

where $\gamma > 0$ influences the rate of weight reduction. For $i = c$ we have $\beta_c = \exp(-\gamma)$ and of course $\beta_1 = 1$.

The disadvantage of notation (1) is the necessity to exchange the order of the elements in $\mathbf{U}$ and the distances $d^2_{\pi_k(i),k}$ which is a time consuming operation in the optimization algorithm. If we denote the inverse function of $\pi_k(i)$ as $\pi_k^{-1}(i)$, then by changing the order of summation over i and by using the identity $\pi_k^{-1}(\pi_k(i)) = i$, criterion (1) may be rewritten as

$$\begin{aligned} J(\mathbf{U}, \mathbf{V}) &= \sum_{i=1}^{c} \sum_{k=1}^{N} \beta_{\pi_k^{-1}(i),k} (u_{\pi_k^{-1}(\pi_k(i)),k})^m \, d^2_{\pi_k^{-1}(\pi_k(i)),k} \\ &= \sum_{i=1}^{c} \sum_{k=1}^{N} \beta_{\pi_k^{-1}(i),k} (u_{i,k})^m \, d^2_{i,k}. \end{aligned} \tag{4}$$

For all data the following equality constraints are proposed

$$\underset{1 \leq k \leq N}{\forall} \sum_{i=1}^{c} \alpha_{\pi_k^{-1}(i),k} u_{i,k} = f_k \tag{5}$$

where as previously for βs, the following conditions are fulfilled $\alpha_{1,k} \geq \alpha_{2,k} \geq \cdots \geq \alpha_{c,k}$. The $\alpha_{i,k} \in [0,1]$ can be considered as the typicality of the kth datum with respect to the ith cluster;

smaller $\alpha_{i,k}$ results in a more atypical data. At the same time, it allows for increasing the repulsive force between group prototypes. This, in turn, should provide better representation of the data structure that may contain clusters that are not always compact and well separated. The parameters αs are derived based on the order of the distances (2); for more detailed description see the further part of this article. The (5) is similar to that used in the conditional fuzzy C-means (CFCM) clustering, introduced by Pedrycz in [38] and generalized in [35]. The data vectors $\mathbf{x}_k$ are clustered under a condition based on some linguistic term defined on y. If this linguistic term is treated as a fuzzy set with a membership function $\mu_C(y)$, then we get a corresponding value $f_k = \mu_C(y_k) \in [0,1]$ for each datum $\mathbf{x}_k$. For example, if the $\mathbf{x}_k$ vectors are records from a medical database and the additional feature is "age," then μ_C can be a membership function of the linguistic term "middle-aged," allowing the CFCM to reveal a structure of the database in the context: "age" IS "middle-aged". Thus, the f_k parameter can be interpreted as an overall (general) typicality of the kth datum.

Parameter $m > 1$ influences the repulsive force between prototypes and thus a fuzziness of the clusters. Larger m results in a lower repulsive force. For $m \to 1^+$, the fuzzy c-means solution becomes a hard one and thus the force is maximal; for $m \to \infty$, the solution is as fuzzy as possible and the force is minimal.

We seek partition matrix $\mathbf{U}$ and prototypes $\mathbf{V}$ by minimizing (1) subject to the equality constraints (5). If $\mathbf{V}$ is fixed, then columns of $\mathbf{U}$ are independent and minimization of (1) can be performed term by term

$$J(\mathbf{U},\mathbf{V}) = \sum_{k=1}^{N} g_k\left(\mathbf{U}\right) \tag{6}$$

where

$$\underset{1\le k\le N}{\forall} g_k\left(\mathbf{U}\right) = \sum_{i=1}^{c} \beta_{\pi_k^{-1}(i),k}\left(u_{i,k}\right)^m \ d_{i,k}^2. \tag{7}$$

The Lagrangian of (7) with the equality constraints (5) is

$$\underset{1\le k\le N}{\forall} G_k\left(\mathbf{U},\lambda_k\right) = \sum_{i=1}^{c} \beta_{\pi_k^{-1}(i),k}\left(u_{i,k}\right)^m \ d_{i,k}^2 - \lambda_k\left[\sum_{i=1}^{c}\alpha_{\pi_k^{-1}(i),k}u_{i,k} - f_k\right] \tag{8}$$

where λ_k is the Lagrange multiplier. Setting the Lagrangian's gradient to zero we obtain

$$\underset{1\le \ell\le N}{\forall} \frac{\partial G_\ell\left(\mathbf{U},\lambda_\ell\right)}{\partial\lambda_\ell} = \sum_{i=1}^{c}\alpha_{\pi_\ell^{-1}(i),\ell}u_{i,\ell} - f_\ell = 0 \tag{9}$$

and

$$\underset{\substack{1\le s\le c\\ 1\le \ell\le N}}{\forall} \frac{\partial G_\ell\left(\mathbf{U},\lambda_\ell\right)}{\partial u_{s,\ell}} = m\beta_{\pi_\ell^{-1}(s),\ell}\left(u_{s,\ell}\right)^{m-1} \ d_{s,\ell}^2 - \lambda_\ell\alpha_{\pi_\ell^{-1}(s),\ell} = 0. \tag{10}$$

From (10) we get

$$u_{s\ell} = \left(\frac{\lambda_\ell}{m}\right)^{\frac{1}{m-1}}\left(\frac{\alpha_{\pi_\ell^{-1}(s),\ell}}{\beta_{\pi_\ell^{-1}(s),\ell}d_{s,\ell}^2}\right)^{\frac{1}{m-1}}. \tag{11}$$

Inserting (11) to (9) we obtain

$$\left(\frac{\lambda_\ell}{m}\right)^{\frac{1}{m-1}}\left[\sum_{i=1}^{c}\alpha_{\pi_\ell^{-1}(i),\ell}\left(\frac{\alpha_{\pi_\ell^{-1}(i),\ell}}{\beta_{\pi_\ell^{-1}(i),\ell}d_{i,\ell}^2}\right)^{\frac{1}{m-1}}\right] = f_\ell. \tag{12}$$

Combination of (11) and (12) yields

$$u_{s\ell} = \frac{f_\ell\left(\frac{\alpha_{\pi_\ell^{-1}(s),\ell}}{\beta_{\pi_\ell^{-1}(s),\ell}d_{s,\ell}^2}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{c}\alpha_{\pi_\ell^{-1}(j),\ell}\left(\frac{\alpha_{\pi_\ell^{-1}(j),\ell}}{\beta_{\pi_\ell^{-1}(j),\ell}d_{j,\ell}^2}\right)^{\frac{1}{m-1}}}. \tag{13}$$

Let us define the following sets:

$$\underset{1\le k\le N}{\forall}\begin{cases}\mathcal{I}_k = \left\{i \,\middle|\, 1\le i\le c;\ d_{i,k}^2 = 0\right\}\\ \widetilde{\mathcal{I}}_k = \{1,2,\ldots,c\}\backslash\mathcal{I}_k.\end{cases} \tag{14}$$

If $\mathcal{I}_k \neq \emptyset$, then the choice of $u_{i,k} = 0$ for $i \in \widetilde{\mathcal{I}}_k$ and $\sum_{i\in\mathcal{I}_k}\alpha_{\pi_k^{-1}(i),k}u_{i,k} = f_k$ results in minimization of criterion function (4), because elements of the partition matrix are zero for nonzero distances, and nonzero for zero distances.

Of course, the αs can be set arbitrarily, but they should satisfy the monotonic condition. Any monotonic (nonincreasing) function in the range [0,1] can be used as well. The typicality criterion can be based on the distance ranking, and the following functions could be applied:

1) exponential: $\alpha_{\pi_k^{-1}(i),k} = \exp(-\gamma\frac{i-1}{c-1})$;
2) power: $\alpha_{\pi_k^{-1}(i),k} = (\frac{c-i}{c-1})^r$, where $r \ge 1$;
3) logarithmic: $\alpha_{\pi_k^{-1}(i),k} = 1 - \frac{\log(i)}{\log(c)}$.

In the above formulas, i stands for the position of the kth datum in the distance ranking. The choice is explained as follows: datum atypicality considered in αs is smaller if its rank in ordered distances to group centers is farther. After experiments according to the methodology described in Section V-D and Appendix, concerning the classification performance, the exponential function was chosen. Thus, in the next part of the article $\alpha_{\pi_k^{-1}(i)} = \beta_{\pi_k^{-1}(i)}$ for all i, k will be used. Thanks to such a choice, (13) takes the following simple form

$$u_{s\ell} = \frac{f_\ell\left(d_{s,\ell}\right)^{\frac{2}{1-m}}}{\sum_{j=1}^{c}\beta_{\pi_\ell^{-1}(j),\ell}\left(d_{j,\ell}\right)^{\frac{2}{1-m}}}. \tag{15}$$

If we assume that the matrix $\mathbf{U}$ and the values of $\beta_{i,k}$ are fixed, then, after setting the gradient of criterion function (1) with respect to a cluster center $\mathbf{v}_s$ to zero, we can obtain

$$\underset{1\le s\le c}{\forall} -2\sum_{k=1}^{N}\beta_{\pi_k^{-1}(s),k}\left(u_{s,k}\right)^m\left(\mathbf{x}_k - \mathbf{v}_s\right) = 0. \tag{16}$$

From (16) we get

$$\underset{1\le s\le c}{\forall}\mathbf{v}_s = \frac{\sum_{k=1}^{N}\beta_{\pi_k^{-1}(s),k}\left(u_{s,k}\right)^m\mathbf{x}_k}{\sum_{k=1}^{N}\beta_{\pi_k^{-1}(s),k}\left(u_{s,k}\right)^m}. \tag{17}$$

The FOCM algorithm consists in alternating finding a partition matrix (15) and prototypes (17). Each time before calculating the partition matrix, the values of $\beta_{\pi_k^{-1}(i),k}$ are updated. In the FCM algorithm and its varieties such alternating calculations are usually started from a random partition matrix or random prototypes. It is well known that such an approach may lead to a local minimum of the criterion function. To avoid this problem, we usually repeat calculations many times with various random initializations of the partition matrix (or the prototypes) [36]. However, in this case we have to adjudicate, which a random realization is the best. For this purpose validity indexes are usually used. This leads to another question, which index among many described in the literature, should be used.

Therefore, in [29] another approach was proposed. To begin the calculations, the initial prototypes were selected from the clustered dataset in a special way, i.e., they were located on the boundary of the convex hull of the data. Such an initialization of prototypes results in faster convergence of the calculations and decreases the possibility of getting stuck in a local minimum of the criterion function. The simple algorithm to find the initial prototypes was based on linguistically defined rules, which are as follows [29]. First, the mean value over the dataset is calculated, and the datum that is most distant from this mean is chosen as the first initial prototype. Next, the distances between this prototype and each datum are calculated. Then, a datum as far as possible from the previous prototype AND as far as possible from the mean value is chosen as the next prototype. Each new prototype should be as far as possible from the previously chosen prototypes AND from the mean value. The algebraic product as the t-norm modeling the AND connection is used. In this article, a slightly different approach based on [39] is applied, where only the areas with high data density are considered while searching for the initial prototypes. It seems that a more natural linguistic definition of initial prototypes could be as follows. The first prototype should be placed in an area of the highest density. The successive prototypes should be placed in areas of the very very high density AND should have very very large distances from the previous prototypes. We applied the same approach to initialize the prototypes of FOCM (see Algorithm 1 in [39]). No modification of the above algorithm was done and the same parameter values were used.

The fuzzy ordered c-means clustering method can be written as the following algorithm.

```
Algorithm I:
1) Fix $c$ $(1 < c < N)$, $m \in (1, \infty)$; $\gamma > 0$. Set the iteration
   index $\ell = 1$.
2) Initialize prototypes $\mathbf{V}^{(1)} = [\mathbf{v}_1^{(1)}, \mathbf{v}_2^{(1)}, \ldots, \mathbf{v}_c^{(1)}]$ using
   Algorithm 1 from [39].
3) Calculate $\beta_{\pi_k^{-1}(i),k}$ using (2) and (3).
4) Update the fuzzy partition matrix $\mathbf{U}^{(\ell)}$ in the $\ell$th iteration
   using (15) with (3) and $\mathbf{V}^{(\ell)}$.
5) Update prototypes $\mathbf{V}^{(\ell+1)}$ using (17) and $\mathbf{U}^{(\ell)}$,
6) If $\|\mathbf{V}^{(\ell+1)} - \mathbf{V}^{(\ell)}\|_F > \xi$ then $\ell \leftarrow \ell + 1$ and go to 3),
   else stop.
```

Remarks: The iterations are stopped as soon as the Frobenius norm of the successive $\mathbf{V}$ matrices difference has fallen below a preset small positive value ξ. In all experiments $\xi = 10^{-4}$ is used. The proposed method can be regarded as a special case of the FCM clustering with the data weighted depending on the sorted distances from the prototypes of the groups. On the other hand, it can also be considered as a robust clustering method, since data which are distant from certain group prototypes are taken into account with a small weights when determining these prototypes. The higher value of γ, the smaller influence of outliers on clusters (their prototypes). Data (classes in classification tasks) should be well represented using the created groups. In particular for imbalanced data, this also applies to classes with a small number of elements. Therefore, the force that repels prototypes from each other should be large. This force is controlled by weighting exponent m and γ parameter. Value of m close to 1.0 and high value of γ results in a greater repulsive force. There is no theoretical basis for selection of m and γ. After experiments concerning the classification performance presented in Appendix, $m = 1.1$ and $\gamma = 6$ were chosen. Separated clustering of the minority and the majority classes may produce the contradictory rules (resulting from overlapping clusters), what, in turn, might reduce the interpretability level. However, in the case of this article, the rules are selected based on criterion of the highest classification performance, even if obtained at the expense of the interpretation ability decrease.

III. Application of Fuzzy Ordered c-Means Clustering to Obtain Premises of If–Then Rules

The binary classifier is designed on the basis of a training set with cardinality L, $\mathcal{T\!R}^{(L)} = \{(\mathbf{x}_1, \theta_1), (\mathbf{x}_2, \theta_2), \ldots, (\mathbf{x}_L, \theta_L)\}$. Each independent datum $\mathbf{x}_i \in \mathbb{R}^n$ has a corresponding dependent datum $\theta_i \in \{+1, -1\}$ which indicates the assignment to one of the two classes: ω_1 (θ_is equal $+1$) or ω_2 (θ_is equal -1). If we define sets $\mathcal{K}_{\omega_1} = \{k | \theta_k = +1\}$, $\mathcal{K}_{\omega_2} = \{k | \theta_k = -1\}$, $\mathcal{T\!R}_{\omega_1}^{(L_1)} = \{(\mathbf{x}_k, \theta_k) | k \in \mathcal{K}_{\omega_1}\}$, $\mathcal{T\!R}_{\omega_2}^{(L_2)} = \{(\mathbf{x}_k, \theta_k) | k \in \mathcal{K}_{\omega_2}\}$, then the training set may be written as $\mathcal{T\!R}^{(L)} = \mathcal{T\!R}_{\omega_1}^{(L_1)} \cup \mathcal{T\!R}_{\omega_2}^{(L_2)}$, where $L_1 + L_2 = L$.

To determine the premises, we perform the FOCM clustering of the patterns from class ω_1 and then from class ω_2. It is assumed that the number of prototypes is equal to c for each class. Indeed, we obtain $2c$ prototypes. The above two clusterings are equivalent to the clustering of the training set $\mathcal{T\!R}^{(L)}$ twice, under different conditions f_k, i.e., first, using $f_k = 1$ for $k \in \mathcal{K}_{\omega_1}$ and $f_k = 0$ for $k \in \mathcal{K}_{\omega_2}$, and then using $f_k = 0$ for $k \in \mathcal{K}_{\omega_1}$ and $f_k = 1$ for $k \in \mathcal{K}_{\omega_2}$. As a result we get: $(\mathbf{U}^{(1)}, \mathbf{V}^{(1)})$ and $(\mathbf{U}^{(2)}, \mathbf{V}^{(2)})$, respectively. The assumption of c prototypes for each class can be intuitively justified. None of the classes is preferred by using a larger number of prototypes. This is especially important for imbalanced data, where a class with small cardinality is not degraded by having fewer prototypes. A greater repulsive force between prototypes results in the wider variety of premises of the rules that describe small classes. In this way both classes are treated equally when determining premises of rules. The choice of rules that are significant for the classification is made subsequently, at the stage of determining the conclusions.

After the partitions of both classes are obtained, each cluster is represented by the Gaussian membership function with the center $\mathbf{v}_i^{(j)}$ and the dispersion $\mathbf{s}_i^{(j)}$, where $i \in \{1, 2, \ldots, c\}$ is a cluster index and $j \in \{1, 2\}$ denotes a class ω_j. Gaussian functions are among the most frequently used membership functions, same as trapezoidal and triangular. The main disadvantage of the trapezoidal function is a large number of parameters (four) to be determined during the learning process. In case of the triangular function, the number of parameters can be limited to two when isosceles triangle is considered. However, such a triangular function can be easily approximated by the Gaussian one. Moreover, the Gaussian functions assure nonzero activation of the classifier rules for any input values combination. The ℓth components of $\mathbf{v}_i^{(j)}$ and $\mathbf{s}_i^{(j)}$ represent, respectively, the location of the center and the dispersion along the ℓth axis in the input space of the data in the ith cluster of the jth class. If we denote the fuzzy partition matrices for ω_1 and ω_2 as $\mathbf{U}^{(1)} = [u_{i,k}^{(1)}]$ and $\mathbf{U}^{(2)} = [u_{i,k}^{(2)}]$, and the prototype matrices as $\mathbf{V}^{(1)} = [\mathbf{v}_1^{(1)}, \mathbf{v}_2^{(1)}, \ldots, \mathbf{v}_c^{(1)}]$ and $\mathbf{V}^{(2)} = [\mathbf{v}_1^{(2)}, \mathbf{v}_2^{(2)}, \ldots, \mathbf{v}_c^{(2)}]$, then the dispersions of the Gaussian membership functions are calculated based on the idea of the fuzzy variance [36] of the ith group

$$\underset{\substack{1\le i\le c\\ j\in\{1,2\}}}{\forall} \left[\mathbf{s}_i^{(j)}\right]^{(\bullet 2)} = \delta \frac{\sum_{k\in\mathcal{K}_{\omega_j}} u_{i,k}^{(j)} \left[\mathbf{x}_k - \mathbf{v}_i^{(j)}\right]^{(\bullet 2)}}{\sum_{k\in\mathcal{K}_{\omega_j}} u_{i,k}^{(j)}} \tag{18}$$

where superscript $(\bullet 2)$ denotes component-by-component squaring, i.e.,

$$\left[\mathbf{x}_k - \mathbf{v}_i^{(j)}\right]^{(\bullet 2)} = \left[\left(x_{k1} - v_{i1}^{(j)}\right)^2, \ldots, \left(x_{kn} - v_{in}^{(j)}\right)^2\right]$$

where n denotes the size of feature vector, and $\delta > 0$ is the scale parameter. The estimation of dispersion is appropriate for normally distributed clusters. However, the distribution of clusters is unknown (it may not be normal), therefore, the scale parameter is adjusted during the learning.

We assume that the fuzzy rule base of classifier consists of a set of the canonical if–then rules with compound premise using the AND operator

$$\mathbf{A}_i^{(j)} = A_{i,1}^{(j)} \times A_{i,2}^{(j)} \times \cdots \times A_{i,n}^{(j)} \tag{19}$$

where $A_{i,\ell}^{(j)}$ is the fuzzy set representing the ℓth component (feature) of the ith cluster in the jth class. $\mathbf{A}_i^{(j)}$ denotes the multidimensional fuzzy set representing the ith cluster in the jth class. If we use the algebraic product as the t-norm corresponding to the AND operator, then the fuzzy set in the premise of the rule can be written as

$$\underset{\substack{1\le i\le c\\ j\in\{1,2\}}}{\forall} \mu_{\mathbf{A}_i^{(j)}}(\mathbf{x}) = \exp\left[-\frac{1}{2}\sum_{\ell=1}^{n}\left(\frac{x_\ell - v_{i,\ell}^{(j)}}{s_{i,\ell}^{(j)}}\right)^2\right]. \tag{20}$$

IV. Application of Least Angle Regression to Obtain Conclusions of If–Then Rules

A set of if–then rules with fuzzy premises and fuzzy conclusions forms a fuzzy rule base, where each rule expresses a piece of knowledge. The canonical Mamdani–Assilian form of the kth fuzzy if–then rule can be put as

$$\text{IF } X_1 \text{ IS } A_{k,1}^{(j)} \text{ AND } \cdots \text{ AND } X_n \text{ IS } A_{k,n}^{(j)} \text{ THEN } Y \text{ IS } B_k^{(j)},$$

where $X_1, \ldots, X_n$ and Y stand for linguistic variables of inputs and output of a fuzzy classifier, and $A_{k,i}^{(j)}, B_k^{(j)}$ are linguistic terms. $A_{k,i}^{(j)}$ is a Gaussian shaped linguistic term for the ith input variable (feature) in the kth rule expressing the knowledge concerning the jth class. $B_k^{(j)}$ is a crisp value (singleton set) in the kth rule of the jth class, i.e., $\mu_{B_k^{(j)}}(y) = \delta_{y,y_k^{(j)}}$, where $y_k^{(j)} \in [-1, +1] \setminus \{0\}$ is a location of the singleton. If $y_k^{(j)}$ is greater than zero, then it corresponds to the class ω_1, otherwise to ω_2. The absolute value of $y_k^{(j)}$ may be treated as a certainty factor of the kth rule.

A set of the above rules for all $k \in \{1, 2, \ldots, c\}$, $j \in \{1, 2\}$ creates a rule base. This rule base is activated by the singleton inputs $\mathbf{x}_\ell = [x_{\ell,1}, x_{\ell,2}, \ldots, x_{\ell,n}]^\top$, where X_1 IS $x_{\ell,1}$, X_2 IS $x_{\ell,2}, \ldots, X_n$ IS $x_{\ell,n}$.

The output value of the fuzzy classifier for the ℓth datum can be put in the form

$$y_{0\ell} = \sum_{j=1}^{2}\sum_{k=1}^{c} \overline{\mu_{\mathbf{A}_k^{(j)}}}(\mathbf{x}_\ell)\, y_k^{(j)} \tag{21}$$

where the normalized activation of the rule is defined as

$$\overline{\mu_{\mathbf{A}_k^{(j)}}}(\mathbf{x}_\ell) = \frac{\mu_{\mathbf{A}_k^{(j)}}(\mathbf{x}_\ell)}{\sum_{j=1}^{2}\sum_{k=1}^{c}\mu_{\mathbf{A}_k^{(j)}}(\mathbf{x}_\ell)}. \tag{22}$$

If we denote $\mathbf{g}(\mathbf{x}_\ell)^\top = [\overline{\mu_{\mathbf{A}_1^{(1)}}}(\mathbf{x}_\ell), \overline{\mu_{\mathbf{A}_2^{(1)}}}(\mathbf{x}_\ell), \ldots, \overline{\mu_{\mathbf{A}_c^{(1)}}}(\mathbf{x}_\ell), \overline{\mu_{\mathbf{A}_1^{(2)}}}(\mathbf{x}_\ell), \overline{\mu_{\mathbf{A}_2^{(2)}}}(\mathbf{x}_\ell), \ldots, \overline{\mu_{\mathbf{A}_c^{(2)}}}(\mathbf{x}_\ell)]$ and $\mathbf{y}^\top = [y_1^{(1)}, y_2^{(j)}, \ldots, y_c^{(1)}, y_1^{(2)}, y_2^{(2)}, \ldots, y_c^{(2)}]$, then (21) can be rewritten in the form

$$y_{0\ell} = \mathcal{F}(\mathbf{x}_\ell) = \mathbf{g}(\mathbf{x}_\ell)^\top \mathbf{y}. \tag{23}$$

To obtain conclusions of the rule base we seek vector $\mathbf{y}$, such that

$$\mathcal{F}(\mathbf{x}_\ell) = \mathbf{g}(\mathbf{x}_\ell)^\top \mathbf{y} \begin{cases} \ge 0, & \mathbf{x}_\ell \in \omega_1 \\ < 0, & \mathbf{x}_\ell \in \omega_2. \end{cases} \tag{24}$$

Taking into account that $\theta_\ell = +1$ iff $\mathbf{x}_\ell \in \omega_1$ and $\theta_\ell = -1$ iff $\mathbf{x}_\ell \in \omega_2$, then (24) can be rewritten in the form $\theta_\ell\, \mathbf{g}(\mathbf{x}_\ell)^\top \mathbf{y} \ge 0$, for $\ell = 1, 2, \ldots, L$. For overlapping classes, which occur almost always, it is impossible to find such an $\mathbf{y}$, that the above conditions are satisfied for all data from the training set. It is possible for perfectly separable case only. But even then some data may lie near the separating curve $\theta_\ell\, \mathbf{g}(\mathbf{x}_\ell)^\top \mathbf{y} = 0$ which leads to a low generalization ability. Thus, a safer approach is to seek vector $\mathbf{y}$, such that

$$\theta_\ell\, \mathbf{g}(\mathbf{x}_\ell)^\top \mathbf{y} \ge \varepsilon;\ \varepsilon > 0 \tag{25}$$

where ε is the margin of separation. Contrary to appearances, the choice of ε is not important. Note that the margin of separation is independent from rescaling (25). If we multiply both sides of (25) by κ, then a new certainty factors vector is $\mathbf{y}^* = \mathbf{y}\kappa$. Thus, we may simply use $\varepsilon = 1$. Solving the inequality system is complicated, thus we usually replace it with the equality system, and we seek such $\mathbf{y}$ that $\theta_\ell \, \mathbf{g}(\mathbf{x}_\ell)^\top \mathbf{y} = 1$ for $\ell = 1, 2, \ldots, L$.

The above equalities can be rewritten in the matrix form $\mathbf{Gy} = \mathbf{1}$, where $\mathbf{1}$ denotes the vector with all entries equal to 1 and $\mathbf{G}$ is the $L \times (2c)$ matrix

$$\mathbf{G}^\top \triangleq \left[\theta_1 \, \mathbf{g}\left(\mathbf{x}_1\right), \theta_2 \, \mathbf{g}\left(\mathbf{x}_2\right), \ldots, \theta_L \, \mathbf{g}\left(\mathbf{x}_L\right)\right]. \tag{26}$$

In minimum squared error procedure of a classifier design, the minimized criterion function takes the form [2], [5], [40]

$$J\left(\mathbf{y}\right) = \sum_{\ell=1}^{L} \left(\theta_\ell \, \mathbf{g}\left(\mathbf{x}_\ell\right)^\top \mathbf{y} - 1\right)^2 = \left(\mathbf{Gy} - \mathbf{1}\right)^\top \left(\mathbf{Gy} - \mathbf{1}\right). \tag{27}$$

The following problem concerning the above minimization arises. How to find a separating hyperplane (described by $\mathbf{y}$) that generalizes well? Some very useful tools for solving this problem are offered by the statistical learning theory. The so-called structural risk minimization induction principle suggests a tradeoff between the quality of the classification on the training set and the complexity of the classifier. So, we should select the classifier with the smallest complexity and the smallest misclassification error on the training set to achieve a good generalization ability. If the complexity is high, then we could minimize misclassification error on the training set down to zero, but the error rate on the testing set might be significant. In this case, the so-called overfitting (overtraining) effect occurs. The above leads to the minimization of the criterion known as ridge regression

$$J\left(\mathbf{y}\right) = \left(\mathbf{Gy} - \mathbf{1}\right)^\top \left(\mathbf{Gy} - \mathbf{1}\right) + \mu \, \mathbf{y}^\top \mathbf{y} \tag{28}$$

where $\mu \geq 0$ controls the tradeoff between the amount up to which errors are tolerated (first term) and the classifier complexity (second term). The ridge regression is an example of shrinkage methods, which shrinks the regression coefficients by imposing a penalty on their size, and μ controls the amount of shrinkage. Even a strong increase of μ does not reset elements of $\mathbf{y}$. They can still be if–then rules (described by $\mathbf{y}$ components), which mutually eliminate each other! The advantage of ridge regression is the simplicity of the solution, by applying quadratic penalty. Regression theory has developed more advanced methods, that force the resetting of some coefficients (the so-called sparse solution). The best-known is least absolute shrinkage and selection operator (LASSO), where L_1 norm penalty $\mu\|\mathbf{y}\|_1$ was applied. This makes the solution nonlinear and it is not possible to present an analytical solution as for the ridge regression. Paper [41] shows the LAR—an extremely effective algorithm for calculation the entire path of solutions as μ varies, with the same computational cost as for the ridge regression algorithm.

The forward stepwise regression builds a model sequentially, adding one variable at a time. In our case, the variable is a column of the matrix $\mathbf{G}$, that corresponds to one if–then rule. At each step the algorithm identifies the best rule to be included in the so-called active set (AS) of rules. After extension of AS, the vector $\mathbf{y}$ is modified to obtain the minimization of the mean square criterion (residual-sum-of-squares) for all active rules. In other words, in each step the number of zeros in $\mathbf{y}$ is reduced by one. The LAR algorithm works similarly, but in each step the new rule is included in the solution only partially. First, the algorithm determines the most correlated rule with residuals $\mathbf{Gy} - \mathbf{1}$ of fuzzy system (best describing recognized classes). However, instead of obtaining the minimization of the squared criterion for this rule, the algorithm increases the value of its weight (y_i, if the ith rule is selected) gradually, until equating the correlation to the residual of another rule. Then the next rule is included in AS, and consequently the number of zeros in $\mathbf{y}$ is reduced by one at a time. Now the weights of both rules are changed at the same time to reduce the residual of the model. The process continues to select all rules, which leads to the traditional method of least squares. The name "least angle" comes from a geometric interpretation—the directions of the weight modification are minimal and equal for all rules in the active set. Details of the algorithm can be found in [41]. After performing the LAR algorithm, we get $\mathbf{y}$ for a changing active set of rules with cardinality from 1 to $2c$: $\mathbf{y}^{[\sigma]}$, where σ denotes number of rules with nonzero weights; $\sigma = 1, 2, \ldots, 2c$ and $\sigma = \sigma_m + \sigma_M$, where σ_m, σ_M denote number of rules describing the minority and majority class, respectively. From all $\mathbf{y}^{[\sigma]}$ we choose the one providing the rules with the greatest generalization ability.

V. Numerical Experiments and Discussion

All experiments were performed on HP Intel Core i7-8700 CPU @ 4.50 GHz with 16 GB RAM, running Windows 10, and MATLAB R2016b environment. The LSVM was obtained from Internet as a set of MATLAB m-files.[1] All algorithms, introduced in the article, were implemented as m-files too. Checking the effectiveness of the developed method consists of five experiments. In each of them, multiple classifiers were compared over multiple datasets. Hypothesis testing was used for the statistical support of the comparison. The methodology described in [42] was applied. In the first stage, the nonparametric Friedman test was used to evaluate the statistical significance of the differences between classifiers. For this purpose, the rank of each classifier was determined for each database, and hence the average rank for each classifier was calculated. Let us denote an average rank of the ith classifier as $R_i = \frac{1}{D} \sum_j r_j^i$, where r_j^i is the rank of the ith of ℓ classifiers on the jth of D datasets. Under hypothesis H_0, that all the classifiers are equivalent (their average ranks are equal), the Friedman statistic [42]

$$\chi_F^2 = \frac{12D}{\ell(\ell+1)} \left[\sum_{i=1}^{\ell} R_i^2 - \frac{\ell(\ell+1)^2}{4}\right] \tag{29}$$

is distributed according to the chi-squared distribution with $\ell - 1$ degrees of freedom. The above statistic is undesirably conservative, therefore Iman and Davenport proposed a better one [42]

$$F_F = \frac{(D-1)\chi_F^2}{D(\ell-1) - \chi_F^2} \tag{30}$$

[1] Accessed on: 3 October 2008. [Online]. Available: http://www.cs.wisc.edu/dmi/lsvm

TABLE I
PARAMETERS OF THE DATABASES USED IN THE FIRST EXPERIMENT

Database Name	Input Dimension	Number of Instances	Imbalance Ratio
Banana	2	5300	1.2
Breast Cancer	9	277	2.5
Diabetis	8	768	1.9
Flare Solar	9	1066	1.2
German	20	1000	2.3
Heart	13	270	1.2
Image	18	2310	1.3
RingNorm	20	7400	1.0
Splice	60	3175	1.0
Thyroid	5	215	2.3
Titanic	3	2201	2.0
TwoNorm	20	7400	1.0
Waveform	21	5000	2.0

which is distributed according to the F-distribution with $\ell - 1$ and $(\ell - 1)(D - 1)$ degrees of freedom. If H_0 is rejected, it means that at least one of the classifiers is significantly different from the others. In the second stage, the *post hoc* Bonferroni-Dunn test is performed by determining the value of the critical distance (CD), which is the minimum difference in the average rank, indicating a significant difference in the performance of the classifiers.

A. First Experiment: Well-Balanced Datasets

In this experiment, 13 well-known benchmark datasets from IDA repository were used.[2] The repository includes 100 (or 20 for `Image` and `Splice`) predefined random partitions into training and testing sets for all datasets (see Table I for the details).

The structure of the experiments was as follows. For each database, multifold cross-validation on 5% of the partitions (into training and testing sets) was used to estimate the following parameters: the number of the if–then rules (c), the centers of the clusters $(\mathbf{v}_i)$, their dispersions $(\mathbf{s}_i)$, and the scale parameter (δ). The scale parameter δ was varied from 0.2 to 2.0 (with the step of 0.1). Parameter c was changed from 2 to 20. The obtained parameters were used in the final multifold cross-validation (100-fold or 20-fold). The average and the standard deviation of the generalization error (overall classification accuracy) were used for comparison with the well-known classifiers, such as: the radial basis function neural network (RBF) [4], [30], the SVM with Gaussian kernel [7], the Regularized AdaBoost (AB$_R$) [15], the kernel Fisher discriminant with the Gaussian kernel (KFD) [16], the LSVM [17], the kernel iteratively reweighted least square (KIRLS) [40] and the fuzzy $(c + p)$-means iteratively reweighted least square (FCPM-IRLS) [29]. The purpose of the first experiment is to compare the generalization ability of the proposed classifier (FOCM-LAR) with the above state-of-the-art classifiers whose results for the abovementioned 13 datasets have already been published. Table II shows the average generalization performance and its standard deviation (confidence interval). For each database, the best results are boldfaced. For the RBF, AB$_R$ and SVM classifiers, the results were taken from [15], and for the KFD from [16]. The parameters δ, c, and σ (the number of rules chosen by the algorithm) obtained for each database by the FOCM-LAR classifier are given in brackets in the last column. The average rank of each algorithm is given in the last row of the table.

The Friedman test calculated with $\ell = 8$ and $D = 13$ based on (29), (30), gives values: $\chi^2_F = 25.97$ and $F_F = 4.79$. These values are higher than the critical values determined for the level of significance 0.05: $\chi^2_{F\text{crit.}} = 14.06$ (for 7 degrees of freedom) and $F_{F\text{crit.}} = 2.12$ (for 7 and 84 degrees of freedom). Therefore, on the basis of both tests H_0 should be rejected. So the statement that at least one of the classifiers significantly differs from the others is true. The CD for the *post hoc* Bonferroni-Dunn test equals 2.11 [42]. The difference between the extreme values of the average rank is 3.84, therefore *post hoc* test is powerful enough to detect significant difference between the classifiers. Analyzing the values of the average rank we can see two groups of algorithms (four in each), which differences of performance are not statistically significant—first group: FCPM-IRLS, KIRLS, FOCM-LAR, and LSVM; second group: KFD, SVM, AB$_R$, and RBF. Thus, the FOCM-LAR classifier introduced in the article is in the group of classifiers with better results, but it is not the best one. However, in this group, it is one of two classifiers with easy interpretation of the knowledge base. From this point of view, its overall accuracy is not statistically worse than FCPM-IRLS, however, the total number of rules for all 13 databases is 252 and 180 for FCPM-IRLS and FOCM-LAR, respectively. In conclusion, the FOCM-LAR classifier leads to statistically comparable results with a simpler knowledge base. The above experiment uses the overall accuracy indicator, which is not suitable for imbalanced datasets, therefore, in the next experiments (imbalanced datasets), the G-mean indicator will be used.

B. Second Experiment: Highly Imbalanced Datasets

In the second experiment 29 well-known benchmark datasets from the University of California-Irvine (UCI) [44] were used.[3] These datasets have a different number of instances, features, and IR (from 1.2 to as many as 129.5). Some of them have instances of two-classes, and others of more than two classes. The multiclass datasets were transformed into two-class datasets (see Table III for the details). The purpose of the second experiment is to compare the generalization ability (using G-mean) of the proposed classifier (FOCM-LAR) with state-of-the-art active learning classifiers for imbalanced data: the random online-sequential weighting (ROW-ELM) [43], the random undersampling extreme learning machine (RUS-ELM) [45], the random oversampling extreme learning machine (ROS-ELM) [45], the bootstrap-based oversampling extreme learning ma-

[2] Accessed on: 3 August 2008. [Online]. Available: http://ida.first.fraunhofer.de/projects/bench

[3] Accessed on: 21 September 2018. [Online]. Available: http://archive.ics.uci.edu/ml

TABLE II
GENERALIZATION ERROR (MEAN ± ST. DEV.) FOR VARIOUS CLASSIFIERS AND DATASETS FROM TABLE I

Database	RBF	AB$_R$	SVM	KFD	LSVM	KIRLS	FCPM-IRLS	FOCM-LAR
	[15]	[15]	[15]	[16]	[40]	[40]	[29]	$(c\|\sigma\|\delta)$
Banana	10.8 ± 0.6	10.9 ± 0.4	11.5 ± 0.7	10.8 ± 0.5	10.3 ± 0.4	10.4 ± 0.4	$\mathbf{10.3 \pm 0.3}$	10.5 ± 0.4 $(9\|18\|1.3)$
Breast Cancer	27.6 ± 4.7	26.5 ± 4.5	26.0 ± 4.7	25.8 ± 4.6	25.1 ± 4.0	24.8 ± 4.2	$\mathbf{21.4 \pm 4.0}$	21.7 ± 0.3 $(8\|11\|0.2)$
Diabetis	24.3 ± 1.9	23.8 ± 1.8	23.5 ± 1.7	23.2 ± 1.6	23.1 ± 1.7	23.0 ± 1.6	$\mathbf{21.7 \pm 1.8}$	22.2 ± 1.6 $(9\|14\|0.2)$
Flare Solar	34.4 ± 2.0	34.2 ± 2.2	32.4 ± 1.8	33.2 ± 1.7	33.6 ± 1.8	32.6 ± 1.6	32.6 ± 1.8	$\mathbf{31.9 \pm 1.9}$ $(5\|9\|0.5)$
German	24.7 ± 2.4	24.3 ± 2.1	23.6 ± 2.1	23.7 ± 2.2	23.6 ± 2.1	23.3 ± 2.2	21.3 ± 2.3	$\mathbf{17.2 \pm 2.3}$ $(20\|34\|0.7)$
Heart	17.6 ± 3.3	16.5 ± 3.5	16.0 ± 3.3	16.1 ± 3.4	15.7 ± 3.3	15.4 ± 3.2	$\mathbf{6.2 \pm 2.3}$	7.0 ± 2.7 $(15\|29\|0.8)$
Image	3.3 ± 0.6	$\mathbf{2.7 \pm 0.6}$	3.0 ± 0.6	4.8 ± 0.6	4.6 ± 0.7	2.9 ± 0.6	12.7 ± 0.6	7.7 ± 0.7 $(15\|27\|1.0)$
RingNorm	1.7 ± 0.2	1.6 ± 0.1	1.7 ± 0.1	$\mathbf{1.5 \pm 0.1}$	9.2 ± 1.4	1.5 ± 0.1	1.8 ± 0.7	14.1 ± 0.3 $(3\|5\|1.6)$
Splice	10.0 ± 1.0	9.5 ± 0.7	10.9 ± 0.7	10.5 ± 0.6	12.0 ± 0.7	10.8 ± 0.6	$\mathbf{6.1 \pm 0.4}$	6.7 ± 0.3 $(7\|12\|1.6)$
Thyroid	4.5 ± 0.3	4.6 ± 2.2	4.8 ± 2.2	4.2 ± 2.1	4.1 ± 2.3	4.2 ± 2.1	$\mathbf{1.5 \pm 1.6}$	4.7 ± 1.9 $(3\|3\|0.2)$
Titanic	23.3 ± 1.3	22.6 ± 1.2	22.4 ± 1.0	23.2 ± 2.0	22.4 ± 1.2	$\mathbf{22.1 \pm 1.7}$	22.4 ± 1.2	23.2 ± 3.3 $(4\|3\|0.4)$
TwoNorm	2.9 ± 0.3	2.7 ± 0.2	3.0 ± 0.2	2.6 ± 0.2	2.4 ± 0.1	2.5 ± 0.2	2.1 ± 0.1	$\mathbf{2.1 \pm 0.0}$ $(3\|5\|1.6)$
Waveform	10.7 ± 1.1	9.8 ± 0.8	9.9 ± 0.4	9.9 ± 0.4	10.3 ± 0.4	9.7 ± 0.4	9.1 ± 0.5	$\mathbf{8.3 \pm 0.1}$ $(6\|10\|0.5)$
Average rank	6.53	5.38	5.31	5.00	4.53	3.08	2.69	3.46

The best results for each database are in boldface. Columns 2–8 contain the results from [15], [16], [29], [40].

TABLE III
DESCRIPTION OF THE DATABASES USED IN THE SECOND EXPERIMENT

Database Name	Input Dimension	Number of Instances	Minority Class of (Instances)	Majority Class (Instances)	Imbalance Ratio
Abalone9	8	4177	class 9 (689)	remaining classes (3488)	6.1
Abalone19	8	4177	class 19 (32)	remaining classes (4145)	129.5
Banknote	4	1372	class 1 (610)	class 0 (762)	1.2
Credit cart	23	30000	class 1 (6636)	class 0 (23364)	3.5
CTGN1vN3	21	1831	class ’NSP3’ (176)	class ’NSP1’ (1655)	9.4
CTGC1	21	2126	class 1 (384)	remaining classes (1742)	4.5
CTGC5	21	2126	class 5 (72)	remaining classes (2054)	28.5
CTGC10	21	2126	class 10 (197)	remaining classes (1929)	9.8
CTGN2	21	2126	class ’NSP2’ (295)	remaining classes (1831)	6.2
CTGN3	21	2126	class ’NSP3’ (176)	remaining classes (1950)	11.1
Haberman	3	306	class 2 (81)	class 1 (225)	2.8
ILPD	10	583	class 2 (167)	class 1 (416)	2.5
Letter A	16	20000	class ’A’ (789)	remaining classes (19211)	24.3
Magic	10	19020	class 1 (6688)	class 2 (12332)	1.8
Mfeat-mor0	6	2000	class 0 (200)	remaining classes (1800)	9.0
Mfeat-mor01	6	2000	class 0 and 1 (400)	remaining classes (1600)	4.0
Mfeat-mor012	6	2000	class 0 to 2 (600)	remaining classes (1400)	2.3
Mfeat-mor0123	6	2000	class 0 to 3 (800)	remaining classes (1200)	1.5
Page-blocks5	10	5473	class 5 (115)	remaining classes (5358)	46.6
Seed2	7	210	class 2 (70)	remaining classes (140)	2.0
Segment grass	19	2310	class ’Grass’ (330)	remaining classes (1980)	6.0
Segment1	19	2310	class 1 (330)	remaining classes (1980)	6.0
Segment12	19	2310	class 1 and 2 (660)	remaining classes (1650)	2.5
Segment123	19	2310	class 1 to 3 (990)	remaining classes (1320)	1.3
Vowel0	13	990	class ’hid’ (90)	remaining classes (900)	10.0
Wilt	5	4839	class 1 (261)	class 2 (4578)	17.5
Yeast-ME1	8	1484	class ’ME1’ (44)	remaining classes (1440)	32.7
Yeast-ME2	8	1484	class ’ME2’ (51)	remaining classes (1433)	28.1
Yeast-ME3	8	1484	class ’ME3’ (163)	remaining classes (1321)	8.1

chine (BootOS-ELM) [45], the active cost sensitive support vector machine (ACS-SVM) [46], the active online-weighted extreme learning machine (AOW-ELM) [43]. The results of these classifiers for considered datasets have been published in [43].

The structure of the experiments was as follows. For each database 100-fold cross-validation was performed. Ten percent of the partitions (into training and testing sets) was used to estimate the parameters: the number of the if–then rules, the centers of the clusters, their dispersions, and the scale parameter.

TABLE IV
COMPARISON OF THE AVERAGE G-MEAN INDEX FOR VARIOUS CLASSIFIERS AND DATASETS FROM TABLE III

Database	ROW-ELM [43]	RUS-ELM [43]	ROS-ELM [43]	BootOS-SVM [43]	ACS-SVM [43]	AOW-ELM [43]	FOCM-LAR $(c\vert\sigma\vert\delta)$
Abalone9	0.6405	0.6209	0.5560	0.5863	0.5388	**0.6445**	0.6413 (19|38|1.5)
Abalone19	0.6115	0.6782	0.5207	0.5607	0.0732	0.5985	**0.7091** (10|20|1.3)
Banknote	0.9885	0.8331	0.9854	0.9970	0.9873	0.9973	**0.9991** (18|35|1.6)
Credit cart	0.6654	0.6487	0.6157	0.6362	0.6616	0.6539	**0.6949** (3| 6|1.6)
CTGN1vN3	0.8848	0.8894	0.8765	0.8878	0.9052	0.6445	**0.9661** (18|27|1.6)
CTGC1	0.8445	0.8378	0.8114	0.8310	0.8684	0.8583	**0.8687** (20|32|1.6)
CTGC5	0.8528	0.8179	0.7360	0.7216	0.8195	0.8835	**0.9133** (10|19|1.6)
CTGC10	0.8955	0.8729	0.8417	0.8627	0.8990	0.9030	**0.9327** (17|32|1.6)
CTGN2	0.8630	0.8582	0.8324	0.8530	0.8709	0.8735	**0.8816** (19|38|1.6)
CTGN3	0.9022	0.8947	0.8837	0.9006	0.9238	0.9079	**0.9479** (18|28|1.3)
Haberman	0.5892	0.5045	0.5173	0.5457	0.5552	0.5714	**0.6763** (4| 7|1.4)
ILPD	0.6266	0.5607	0.5482	0.5957	0.6206	0.6303	**0.7126** (15| 8|1.1)
Letter A	0.9613	0.8920	0.9397	0.9506	**0.9738**	0.9707	0.9626 (7| 8|1.4)
Magic	0.8178	0.0000	0.7927	0.8123	**0.8344**	0.7703	0.7936 (15|29|1.6)
Mfeat-mor0	0.9899	0.9590	0.9898	0.9907	0.9901	0.9903	**0.9928** (3| 2|1.5)
Mfeat-mor01	0.9738	0.9030	0.9660	0.9708	0.9692	0.9755	**0.9767** (8| 6|0.2)
Mfeat-mor012	0.8981	0.8517	0.8684	0.8812	0.8939	0.9059	**0.9244** (12|16|0.2)
Mfeat-mor0123	0.8247	0.6990	0.7767	0.7994	0.8368	0.8285	**0.8778** (14|27|1.3)
Page-blocks5	0.8893	0.8363	0.8328	0.8294	0.8402	0.8739	**0.9373** (9| 9|0.8)
Seed2	0.9750	0.9044	**0.9833**	0.9829	0.9621	0.9810	0.9680 (2| 3|0.2)
Segment grass	0.9977	0.9746	0.9974	0.9998	**1.0000**	0.9962	0.9978 (18|18|0.9)
Segment1	0.9892	0.9565	0.9897	0.9873	**0.9929**	0.9924	0.9705 (18|26|1.6)
Segment12	0.9921	0.9731	0.9899	0.9904	**0.9958**	0.9932	0.9759 (18|36|1.6)
Segment123	0.9320	0.9137	0.9197	0.9243	**0.9629**	0.9403	0.9591 (20|40|1.6)
Vowel0	0.9651	0.9338	0.9413	0.9860	0.9788	0.9857	**0.9918** (6| 6|0.2)
Wilt	0.9224	0.8258	0.8342	0.9005	**0.9512**	0.9316	0.9224 (20|29|0.9)
Yeast-ME1	0.9326	0.9201	0.8732	0.9069	0.9255	0.9403	**0.9629** (2| 2|0.5)
Yeast-ME2	0.9313	0.9178	0.8940	0.9336	0.9146	**0.9421**	0.8592 (15| 3|1.3)
Yeast-ME3	0.9183	0.8711	0.8934	0.8985	0.9105	0.9188	**0.9256** (9|10|0.2)
Average rank	3.45	5.93	5.86	4.55	3.27	2.83	2.10

The best results for each database are in boldface. Columns 2–7 contain the results from [43].

The obtained parameters were used in the final 100-fold cross-validation. The average of G-mean was used for comparison with the well-known classifiers (see Table IV). For each database, the best results are boldfaced. The results of ROW-ELM, RUS-ELM, ROS-ELM, BootOS-ELM, ACS-SVM, and AOW-ELM are taken from [43]. The parameters δ, c, and σ obtained for each database by the FOCM-LAR classifier are given in brackets, in the last column. The average rank of each algorithm is given in the last row of the table.

The Friedman test calculated with $\ell = 7$ and $D = 29$ based on (29), (30), gives values: $\chi^2_F = 82.16$ and $F_F = 25.04$. They are higher than the critical values determined for the level of significance 0.05: $\chi^2_{F\text{crit.}} = 12.59$ (for 6 degrees of freedom) and $F_{F\text{crit.}} = 2.15$ (for 6 and 168 degrees of freedom). On the basis of both tests, H_0 should be rejected and thus, at least one of the classifiers is significantly different from the others. The *CD* for the *post hoc* Bonferroni-Dunn test is 1.50 [42]. The difference between the extreme values of the average rank is 3.83, therefore *post hoc* test is powerful enough to detect significant difference between the classifiers. Based on the values of the average rank and *CD* value we can see that the classifier proposed in the article has the best position in the ranking, however, the differences from AOW-ELM, ACS-SVM, ROW-ELM classifiers are not statistically significant. However, the classifier proposed (FOCM-LAR) achieves comparable results to the best of active learning methods. Additionally, its knowledge base is interpretable.

C. Third Experiment: Moderately Imbalanced Datasets

In the third experiment 17 well-known benchmark datasets from the University of California-Irvine (UCI)[4] [44] and three datasets named `OCR`, `Phoneme`, and `Texture` from UCL Neural Network Group, The ELENA Project[5] were used. These datasets have a different number of instances, features, and IR (from 1.8 to 22.7). Some of them have instances of two-classes, and others were transformed into two-class datasets

[4] Accessed on: 21 September 2018. [Online]. Available: http://archive.ics.uci.edu/ml

[5] Accessed on: 12 October 2018. [Online]. Available: https://www.elen.ucl.ac.be/neural nets/Research/Projects/ELENA/databases/REAL/

TABLE V
DESCRIPTION OF THE DATABASES USED IN THE THIRD EXPERIMENT

Database Name	Input Dimension	Number of Instances	Minority Class of (Instances)	Majority Class (Instances)	Imbalance Ratio
Abalone18v9	8	731	class 18 (42)	class 9 (689)	16.4
Breast Cancer Original	9	683	class 'Malignant' (239)	class 'Benign' (444)	1.8
Breast Tissue	9	106	classes 'CAR' and 'FAD' (36)	remaining classes (70)	1.9
CTG34	21	2126	classes 3 and 4 (134)	class 0 (1992)	14.9
Ecoli	7	336	class 'im' (77)	remaining classes (259)	3.4
Glass	9	214	classes 5 to 7 (51)	remaining classes (163)	3.2
Libra	90	360	classes 1 to 3 (72)	remaining classes (288)	4.0
OCR	64	3823	class 0 (376)	remaining classes (3447)	9.2
Pageblocks	10	5473	classes 'Graphic', 'Vert', 'Picture' (231)	remaining classes (5242)	22.7
Phoneme	5	5404	class 1 (1586)	remaining classes (3818)	2.4
Pima	8	768	class 1 (268)	class 0 (500)	1.9
Robot	24	5456	class 'slight-left' and 'slight-right' (1154)	remaining classes (4302)	3.7
Satimage	36	6435	classes 2,4,5 (2036)	remaining classes (4399)	2.2
Segment Grass	19	2310	class 'Grass' (330)	remaining classes (1980)	6.0
Semeion4	256	1593	class '4' (161)	remaining classes (1432)	8.9
StatLandSat4	36	4435	class 4 (415)	remaining classes (4020)	9.7
Texture	40	5500	classes 2 to 4 (1500)	remaining classes (4000)	2.7
Vehicle	18	846	class 1 (218)	remaining classes (628)	2.9
Wine	13	178	class 3 (48)	remaining classes (130)	2.7
Yeast	8	1448	classes 'ME3','ME2','EXC','VAC','POX', 'ERL' (304)	remaining classes (1180)	3.9

(see Table V for the details). The purpose of the third experiment is to compare the generalization ability (using G-mean) of the proposed classifier (FOCM-LAR) with state-of-the-art minority oversampling technique-based classifiers for imbalanced data: the synthetic minority oversampling technique (SMOTE) [47], the adaptive synthetic sampling technique (ADASYN) [48], the ranked minority oversampling (RAMO) [49], and the majority weighted minority oversampling technique (MWMOTE) [50]. The results for all the above-mentioned 20 datasets and classifiers have been published in [50]. We performed similar experiments as previously and the results are shown in Table VI. For each database, the best results are boldfaced.

The results of SMOTE, ADASYN, RAMO, and MWMOTE are taken from [50]. The parameters δ, c, and σ obtained for each database and the average rank of each algorithm are given as previously. The Friedman test calculated with $\ell = 5$ and $D = 20$ based on (29), (30) gives values: $\chi^2_F = 30.92$ and $F_F = 11.96$. These values are higher than the critical values determined for the level of significance 0.05: $\chi^2_{F\text{crit.}} = 9.48$ (for 4 degrees of freedom) and $F_{F\text{crit.}} = 2.49$ (for 4 and 76 degrees of freedom). Therefore, on the basis of both tests, H_0 should be rejected and at least one of the classifiers is significantly different from the others. The *CD* for the *post hoc* Bonferroni–Dunn test equals 1.25 [42]. The difference between the extreme values of the average rank is 2.65, therefore *post hoc* test is powerful enough to detect significant difference between the classifiers. Analyzing the values of the average rank we can see that the classifier proposed in the article achieves statistically significantly better results with respect to state-of-the-art minority oversampling technique-based classifiers for imbalanced data.

D. Fourth Experiment: Imbalanced Cardiotocographic Signals Datasets

In the experiment we used the CTU-UHB dataset [52], [53]. It contains 552 CardioTocoGraphic (CTG) recordings with associated neonatal outcome attributes, which were collected in the University Hospital in Brno, Czechia. Since only raw signals were available, we calculated a set of features that quantitatively describe them in the time domain using the computerized fetal monitoring system MONAKO [54]. Twelve features that are essential when evaluating the fetal state were selected by an expert clinician [51]. The analyzed subdatasets are as follows: 1) due to the output parameter—Apgar score (AS), birth weight (BW), and pH (pH); 2) due to the interpretation of suspect (ambiguous) casessuspect as normal (SAN) and suspect as abnormal (SAA); 3) due to the presence of suspect cases in the learning setsall data (AD) in the training set and suspect cases rejected (SR). (see [51] for details). The purpose of the experiment is to compare the generalization ability (using G-mean) of the proposed classifier (FOCM-LAR) and the best performing classifiers for cardiotocographic datasets (whose results obtained for all the above-mentioned six subdatasets have been published in [51]).

We performed similar experiments, as previously mentioned. The average G-means were used for comparison with the classifiers, such as: the fuzzy C-means iteratively reweighted least squares (FCM-IRLS) [29], the fuzzy $(c + p)$-means iteratively reweighted least squares (FCPM-IRLS) [29], the LSVM [17], and the fuzzy classifier based on clustering with pairs of ε-hyperballs (CPP$^{\varepsilon}_{\text{ST}}$) [51].

Table VII summarizes the obtained results. For each database, the best results are boldfaced. The results of FCM-IRLS, FCPM-IRLS, LSVM, and CPP$^{\varepsilon}_{\text{ST}}$ are taken from [51]. The parameters δ, c, σ_m, and σ_M obtained for each database and the average

TABLE VI
COMPARISON OF THE GENERALIZATION ABILITY USING G-MEAN INDEX (MEAN ± ST. DEV.) FOR VARIOUS CLASSIFIERS AND DATASETS FROM TABLE V

Database	SMOTE [50]	ADASYN [50]	RAMO [50]	MWMOTE [50]	FOCM-LAR $(c\vert\sigma\vert\delta)$
Abalone18v9	0.62356	0.44381	0.58385	0.51451	**0.72002** $(3\vert 4\vert 0.2)$
Breast Cancer Original	0.97438	0.97454	**0.97509**	0.97432	0.97242 $(13\vert 25\vert 1.5)$
Breast Tissue	0.71560	0.73618	0.71236	0.77634	**0.82521** $(4\vert 7\vert 1.6)$
CTG34	0.70765	0.70256	0.67389	0.72320	**0.92187** $(17\vert 21\vert 1.3)$
Ecoli	0.87041	0.87213	0.86449	0.85931	**0.90618** $(4\vert 6\vert 0.8)$
Glass	0.87774	0.92462	0.90413	**0.93611**	0.92739 $(6\vert 9\vert 0.2)$
Libra	0.88706	0.91104	0.88728	0.95059	**0.98238** $(7\vert 7\vert 1.4)$
OCR	0.99255	0.98968	0.98841	0.98861	**0.99777** $(20\vert 20\vert 0.3)$
Pageblocks	**0.99032**	0.98545	0.97554	0.97259	0.94599 $(15\vert 16\vert 0.5)$
Phoneme	0.75935	0.74213	0.73113	0.76618	**0.83231** $(20\vert 38\vert 1.1)$
Pima	0.73159	0.74335	0.70707	0.74088	**0.79708** $(19\vert 38\vert 1.0)$
Robot	0.80532	0.79437	0.73985	0.80073	**0.89880** $(20\vert 40\vert 1.6)$
Satimage	0.81077	0.79614	0.71040	0.81800	**0.88166** $(18\vert 31\vert 1.6)$
Segment Grass	0.77089	0.74559	0.76765	0.79108	**0.99785** $(18\vert 18\vert 0.9)$
Semeion4	0.91312	0.92025	0.89846	0.92444	**0.98299** $(11\vert 11\vert 1.2)$
StatLandSat4	0.68017	0.69035	0.59787	0.73219	**0.86845** $(19\vert 27\vert 1.6)$
Texture	0.93126	0.92088	0.91116	0.93872	**0.97917** $(20\vert 27\vert 0.3)$
Vehicle	0.96321	0.93200	0.96702	0.96610	**0.98432** $(11\vert 17\vert 1.5)$
Wine	0.97483	0.99215	0.96928	0.99215	**0.99621** $(4\vert 4\vert 0.2)$
Yeast	0.82486	0.83142	0.84089	0.83093	**0.87627** $(11\vert 20\vert 0.4)$
Average rank	3.35	3.30	4.10	2.80	1.45

The best results for each database are in boldface. Columns 2–5 contain the results from [50].

TABLE VII
COMPARISON OF THE GENERALIZATION ABILITY USING G-MEAN INDEX (MEAN ± ST. DEV.)
FOR CARDIOTOCOGRAPHIC SIGNALS SUBDATASETS (THE FOURTH EXPERIMENT)

Database	FCM-IRLS [51]	FCPM-IRLS [51]	LSVM [51]	CPP$^{\varepsilon}_{\mathrm{ST}}$ [51]	FOCM-LAR $(c\vert\sigma_m+\sigma_M\vert\delta)$
AS-SAN-AD	0.5891 ± 0.1365	0.6713 ± 0.1700	0.5186 ± 0.0234	0.7154 ± 0.1185	$\mathbf{0.8103 \pm 0.0851}$ $(10\vert 4+10\vert 1.2)$
BW-SAN-AD	0.6430 ± 0.1364	0.5831 ± 0.1582	0.4654 ± 0.0335	0.6739 ± 0.1009	$\mathbf{0.8399 \pm 0.0748}$ $(5\vert 2+6\vert 1.1)$
pH-SAN-AD	0.6420 ± 0.0694	0.6009 ± 0.0704	0.4659 ± 0.0735	0.6639 ± 0.0745	$\mathbf{0.8131 \pm 0.0620}$ $(14\vert 9+18\vert 1.4)$
AS-SAA-AD	0.6250 ± 0.0712	0.5782 ± 0.0611	0.4864 ± 0.0224	0.6679 ± 0.0661	$\mathbf{0.7707 \pm 0.0540}$ $(20\vert 12+22\vert 1.4)$
BW-SAA-AD	0.5060 ± 0.0764	0.5413 ± 0.0845	0.4498 ± 0.0278	0.6593 ± 0.0872	$\mathbf{0.8023 \pm 0.0668}$ $(13\vert 3+14\vert 1.1)$
pH-SAA-AD	0.6527 ± 0.0392	0.6039 ± 0.0312	0.5232 ± 0.0334	0.6604 ± 0.0367	$\mathbf{0.7603 \pm 0.0264}$ $(19\vert 15+20\vert 0.8)$
AS-SAA-SR	0.3885 ± 0.0882	0.3511 ± 0.0927	0.4912 ± 0.0229	0.4295 ± 0.0780	$\mathbf{0.6460 \pm 0.0609}$ $(10\vert 2+8\vert 1.6)$
BW-SAA-SR	0.4327 ± 0.0997	0.4127 ± 0.0637	0.4494 ± 0.0283	0.4923 ± 0.0941	$\mathbf{0.6969 \pm 0.0713}$ $(11\vert 2+9\vert 1.6)$
pH-SAA-SR	0.4609 ± 0.0532	0.3995 ± 0.0537	0.4534 ± 0.0475	0.4923 ± 0.0526	$\mathbf{0.6790 \pm 0.0344}$ $(16\vert 12+18\vert 0.8)$
AS-SAN-SR	0.6014 ± 0.1127	0.5573 ± 0.1081	0.5108 ± 0.0231	0.6685 ± 0.0985	$\mathbf{0.8076 \pm 0.0865}$ $(10\vert 5+12\vert 1.2)$
BW-SAN-SR	0.6765 ± 0.1169	0.5698 ± 0.1364	0.4642 ± 0.0318	0.7303 ± 0.1054	$\mathbf{0.8410 \pm 0.0745}$ $(5\vert 2+5\vert 1.1)$
pH-SAN-SR	0.6700 ± 0.0776	0.6680 ± 0.0933	0.4956 ± 0.0741	0.6898 ± 0.0779	$\mathbf{0.8714 \pm 0.0677}$ $(17\vert 2+16\vert 1.2)$
Average rank	3.33	4.08	4.50	2.08	1.00

The best results for each database are in boldface. Columns 2–5 contain the results from [51].

rank of each algorithm are given. The Friedman test calculated with $\ell = 5$ and $D = 12$ based on (29), (30) provides values: $\chi^2_F = 39.89$ and $F_F = 54.15$, which are higher than the critical values determined for the level of significance 0.05: $\chi^2_{F\text{crit.}} = 9.48$ (for 4 degrees of freedom) and $F_{F\text{crit.}} = 2.58$ (for 4 and 44 degrees of freedom). On the basis of both tests at least one of the classifiers is significantly different from the others. The *CD* for the *post hoc* Bonferroni–Dunn test is 1.61 [42]. The difference between the extreme values of the average rank is 3.5, therefore *post hoc* test is powerful enough to detect significant difference between the classifiers. Taking into account the values of the average rank and *CD* value we can see that the classifier proposed in the article holds the best position in the ranking and the differences between FOCM-LAR and CPP$^{\varepsilon}_{\mathrm{ST}}$ classifiers are not statistically significant. The final rule numbers selected by the LAR algorithm for the minority (σ_m) and majority (σ_M) classes are different. For each of the considered databases, the highest generalization ability was obtained using a smaller number of conditional rules for the minority class that represents rare cases of abnormal fetal condition.

TABLE VIII
GENERALIZATION ABILITY OF THE FOCM-LAR FOR DATASETS USED IN [61]

DIA	AUS	BRE	ADU	MUS
0.8092	0.8672	0.9662	0.8296	0.9138
29	21	25	31	33
MAG	SON	VOT	WBP	LIV
0.7937	0.6279	0.9364	0.7814	0.6841
30	37	12	28	18
SEI	SPE	BAL	PAG	MON
0.9327	0.7813	0.8519	0.9698	0.6926
29	18	10	11	15

Each cell contains dataset name, the generalization ability and number of obtained rules.

E. Fifth Experiment: Comparison to Traditional Fuzzy Classifiers and Deep Learning Methods

The purpose of the fifth experiment is to compare the generalization ability of the proposed classifier (FOCM-LAR) with the traditional fuzzy classifiers: the Bayes classifier defined by mixture of Gaussian density (BCMG) [55], the Classifier with axes-parallel hyperellipsoids (CA-PH) [56], the Takagi–Sugeno fuzzy models (TSFM) [57], the fuzzy clustering techniques for fuzzy model identification (FCTFMI) [58], the neuro-fuzzy classifier (NFC) [59], and the supervised fuzzy clustering for rule extraction (SFCRE) [60]. We performed similar experiments as previously for all the abovementioned 74 datasets. The average rank of each algorithm is as follows: BCMG – 4.14, CA-PH – 4.96, TSFM – 3.73, FCTFMI – 5.06, NFC – 6.39, SFCRE – 2.67, and FOCM-LAR – 1.05. The Friedman test provides values: $\chi_F^2 = 290.52$ and $F_F = 138.18$, which are higher than the critical values determined for the level of significance 0.05: $\chi^2_{F\text{crit.}} = 12.59$ and $F_{F\text{crit.}} = 2.11$. On the basis of both tests at least one of the classifiers is significantly different from the others. The *CD* for the *post hoc* Bonferroni-Dunn test is 0.94 [42]. Taking into account the values of the average rank and *CD* value we can see that the classifier proposed in the article holds the best position in the ranking and the differences between FOCM-LAR and others classifiers are statistically significant.

The proposed method was also compared to deep TSK fuzzy classifier proposed in [61]. For this purpose, the generalization ability of the FOCM-LAR for databases used in [61] was calculated and is presented in Table VIII. Based on this table and [61, Table III], the average ranks of the following algorithms can be determined: three TSK classifiers: zero-order-TSK-FC (7.73), L2-TSK-FC (7.73), FS-FCSVM (7.80); deep learning-based classifiers: DBN (5.26), HID-TSK-FC (3.33); rule-based fuzzy classifiers: FH-GBML-C (5.60), GFS-AdaBoost-C (7.46); hierarchical fuzzy classifier: t-HFC (7.46); SVM classifiers: LIBSVM-linear (6.60), LIBSVM-Gaussian (5.13); and FOCM-LAR (1.86). The Friedman test provides values: $\chi_F^2 = 53.31$ and $F_F = 7.71$, which are higher than the critical values determined for the level of significance 0.05: $\chi^2_{F\text{crit.}} = 18.30$ and $F_{F\text{crit.}} = 1.90$. On the basis of both tests at least one of the classifiers is significantly different from the others. The *CD* for the *post hoc* Bonferroni-Dunn test is 2.43 [42]. We can see that the classifier proposed in the artcle holds the best position in the ranking. The differences between FOCM-LAR and HID-TSK-FC classifiers are not statistically significant, and the numbers of rules are comparable. All other classifiers lead to statistically worse results.

TABLE IX
MEAN GENERALIZATION ABILITY FOR THE FUZZY CLASSIFIER WITH VARIOUS CLUSTERING METHODS

Method	FOCM	FCM1	FCM2	FCM3	FCPM
G-Mean	0.7782	0.6067	0.5712	0.5968	0.6813

Finally, the following experiment was carried out: in the proposed classifier design procedure, the FOCM clustering was replaced with other methods known from the literature: FCM1—fuzzy C-means with initialization from [39]; FCM2—fuzzy C-means with random initialization (50 trails); FCM3—fuzzy C-means with initialization from [29], and FCPM—fuzzy $(C + P)$-means. Table IX shows the mean generalization ability (G-mean index) obtained with the above clustering methods for datasets from the fourth experiment. Analyzing this table, we see that the method proposed in this article derives its strength from both FOCM clustering and the prototypes initialization method drawn from the work [39].

VI. CONCLUSION

In this article, we presented the FOCM-LAR, a new binary classifier design method for imbalanced classes. The method is based on a new fuzzy ordered c-means clustering that increases repulsive force between group prototypes. A better representation of the data class, in particular for classes with small cardinality in the training set (imbalanced data) is obtained through a more local impact of data on created groups via ordering and weighting distances from cluster prototypes. A special initialization of this clustering was also introduced. The FOCM clustering method is used to determine premises of the classifier if–then rules. Using the clustering independently for each class (the majority and the minority), a redundant number of rules was obtained, which was subsequently reduced by means of the LAR algorithm.

An extensive experimental analysis on 89 benchmark datasets shows that the proposed classifier design method is a good alternative, in terms of both generalization ability (measured by G-mean) and interpretability, to state-of-the-art traditional algorithms, as well as to the active learning methods and at least, to state-of-the-art minority oversampling technique classifiers. Our research also included imbalanced cardiotocographic signals datasets. The statistical comparison of the abovementioned classifiers was performed using Friedman, Iman-Davenport and the *post hoc* Bonferroni–Dunn tests. The comparison between the proposed FOCM-LAR and the reference classifiers shows that the FOCM-LAR has the best position in the ranking, although the differences of the performance are not always statistically significant. However, the FOCM-LAR provides a knowledge base that is easily interpretable for humans.

TABLE X
GENERALIZATION ABILITY FOR VARIOUS VALUES OF m AND γ

	γ				
m	0.5	2	4	6	8
1.1	0.8162	0.8173	0.8182	0.8211	0.8001
1.5	0.7477	0.7371	0.7320	0.7288	0.7250
2.0	0.6511	0.6691	0.6862	0.6620	0.6495

APPENDIX

Experiments were carried out according to the methodology described in Section V-D for the following datasets: AS-SAN-AD, BW-SAN-AD, and pH-SAN-AD and clustering parameters with values: $m \in \{1.1, 1.5, 2.0\}, \gamma \in \{0.5, 2, 4, 6, 8\}$. The mean generalization ability using G-mean index for the above values of m and γ are presented in Table X. The highest generalization ability is noticed for $m = 1.1$ and $\gamma = 6$, and such values were used in all experiments.

ACKNOWLEDGMENT

The authors would like to thank anonymous referees for their constructive comments that have helped to improve the quality and presentation of this article.

REFERENCES

[1] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York, NY, USA: Wiley, 1973.

[2] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2001.

[3] J. Tou and R. Gonzalez, *Pattern Recognition Principles*. London, U.K.: Addison-Wesley, 1974.

[4] B. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge University Press, 1996.

[5] A. Webb, *Statistical Pattern Recognition*. London, U.K.: Arnold, 1999.

[6] B. Scholkopf and A. Smola, *Learning With Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. London, U.K.: The MIT Press, 2002.

[7] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[8] L. Kuncheva, *Fuzzy Classifier Design*. Heidelberg, Germany: Physica-Verlag, Springer-Verlag Comp., 2000.

[9] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1367–1381, Jun. 2019.

[10] J. Mathew, C. Pang, M. Luo, and W. Hoe, "Classification of imbalanced data by oversampling in kernel space of support vector machines," vol. 29, no. 9, pp. 465–476, Sep. 2018.

[11] M. Gao, X. Hong, and C. Harris, "Construction of neurofuzzy models for imbalanced data classification," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 6, pp. 1472–1488, Dec. 2014.

[12] S. Liu, J. Zhang, Y. Xiang, and W. Zhou, "Fuzzy-based information decomposition for incomplete and imbalanced data learning," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1476–1490, Dec. 2017.

[13] E. Ramentol *et al.*, "IFROWANN: Imbalanced fuzzy-rough ordered weighted average nearest neighbor classification," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1622–1637, Oct. 2015.

[14] J.-H. Ri, L. Liu, Y. Liu, H. Wu, W. Huang, and H. Kim, "Optimal weighted extreme learning machine for imbalanced learning with differential evolution [Research Frontier]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 32–47, Aug. 2018.

[15] G. Ratsch, T. Onoda, and K.-R. Muller, "Soft margins for AdaBoost," *Mach. Learn.*, vol. 42, no. 3, pp. 287–320, 2001.

[16] S. Mika, G. Ratsch, J. Weston, S. Scholkopf, and K.-R. Muller, "Fisher discriminant analysis with kernels," in *Proc. Neural Netw. Signal Process. IX.*, New York, NY, USA: IEEE Press, 1999, pp. 41–48.

[17] O. Mangasarian and D. Musicant, "Lagrangian support vector machines," *J. Mach. Learn. Res.*, vol. 1, no. 1, pp. 161–177, 2001.

[18] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[19] I. Tsang, J. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 363–392, 2005.

[20] I. W.-H. Tsang, J. T.-Y. Kwok, and J. Zurada, "Generalized core vector machines," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1126–1142, Sep. 2006.

[21] I. Tsang, A. Kocsor, and J. T.-Y. Kwok, "Large-scale maximum margin discriminant analysis using core vector machines," *IEEE Trans. Neural Netw.*, vol. 19, no. 4, pp. 610–623, Apr. 2008.

[22] P. P. Angelov and X. Zhou, "Evolving fuzzy-rule-based classifiers from data streams," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1462–1475, Dec. 2008.

[23] E. Czogala and J. Leski, *Fuzzy and Neuro-Fuzzy Intell. Syst.*, Heidelberg, Germany: Physica-Verlag, Springer-Verlag Comp., 2000.

[24] J. Leski, "An ε-margin nonlinear classifier based on if-then rules," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 34, no. 1, pp. 68–76, Feb. 2004.

[25] E. Lughofer and O. Buchtala, "Reliable all-pairs evolving fuzzy classifiers," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 4, pp. 625–641, Aug. 2013.

[26] 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.

[27] S.-M. Zhou and J. Gan, "Constructing L2-SVM-based fuzzy classifiers in high-dimensional space with automatic model selection and fuzzy rule ranking," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 3, pp. 398–409, Jun. 2007.

[28] J. Leski, "TSK-fuzzy modeling based on ε-insensitive learning," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 2, pp. 181–193, Apr. 2005.

[29] J. Leski, "Fuzzy $(c + p)$-means clustering and its application to a fuzzy rule-based classifier: Towards good generalization and good interpretability," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 802–812, Aug. 2015.

[30] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice Hall, 1999.

[31] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[32] W. Pedrycz, "Fuzzy set technology in knowledge discovery," *Fuzzy Sets Syst.*, vol. 98, no. 2, pp. 279–290, 1998.

[33] W. Pedrycz,"Cluster-centric fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 6, pp. 1585–1597, Dec. 2014.

[34] X. Gu, F.-L. Chung, H. Ishibuchi, and S. Wang, "Imbalanced TSK fuzzy classifier by cross-class Bayesian fuzzy clustering and imbalance learning," *IEEE Trans. Fuzzy Syst.*, vol. 47, no. 8, pp. 2005–2020, Aug. 2017.

[35] J. Leski, "Generalized weighted conditional fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 709–715, Dec. 2003.

[36] J. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms.*, New York, NY, USA: Plenum Press, 1982.

[37] J. Leski, "Fuzzy c-ordered-means clustering," *Fuzzy Sets Syst.*, vol. 286, no. 1, pp. 114–133, 2016.

[38] W. Pedrycz, "Conditional fuzzy clustering in the design of radial basis function neural network," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 601–612, Jul. 1998.

[39] J. Leski and M. Kotas, "Linguistically defined clustering of data," *Int. J. Appl. Math. Comput. Sci.*, vol. 28, no. 3, pp. 545–557, 2018. [Online]. Available: https://www.degruyter.com/downloadpdf/j/amcs.2018.28.issue-3/amcs-2018-0042/amcs-2018-0042.pdf

[40] J. Leski, "Iteratively reweighted least squares classifier and its ℓ_2- and ℓ_1-regularized kernel versions," *Bull. Polish Acad.: Tech.*, vol. 58, no. 1, pp. 171–182, 2010.

[41] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression (with discussion)," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[42] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[43] H. Yu, X. Yang, S. Zheng, and C. Sun, "Active learning from imbalanced data: A solution of online weighted extreme learning machine," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1088–1103, Apr. 2019.

[44] D. Dua and K. Taniskidou., *UCI Machine Learning Repository*. Irvine, CA, USA: School Inf. Comput. Sci., Univ. California, 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[45] J. Zhou and E. Hovy, “Active learning for word sense disambiguation with methods for addressing the class imbalance problem,” in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2007, pp. 783–790.

[46] M. Bloodgood and K. Vijay-Shanker, “Taking into account the differences between actively and passively acquired data: The case of active learning with support vector machines for imbalanced datasets,” in *Proc. Human Lang. Technol.*, 2009, pp. 137–140.

[47] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, “SMOTE: Synthetic minority oversampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[48] H. He, Y. Bai, E. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *Proc. Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[49] S. Chen, H. He, and E. Garcia, “RAMOBoost: Ranked minority oversampling in boosting,” *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1624–1642, Oct. 2010.

[50] S. Barua, M. Islam, X. Yao, and K. Murase, “MWMOTE—Majority weighted minority oversampling technique for imbalanced data set learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–424, Feb. 2014.

[51] M. Jezewski, R. Czabanski, J. Leski, and J. Jezewski, “Fuzzy classifier based on clustering with pairs of ε-hyperballs and its application to support fetal state assessment,” *Expert Syst. Appl.*, vol. 118, no. 1, pp. 109–126, 2019.

[52] V. Chudacek, J. Spilka, M. Bursa, P. Janku, L. Hruban, and M. Huptych, “Open access intrapartum CTG database,” *BMC Pregnancy Childbirth*, vol. 14, no. 1, pp. 1–12, 2014.

[53] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. Ivanov, and R. Mark, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[54] J. Jezewski, J. Wrobel, K. Horoba, T. Kupka, and A. Matonia, “Centralised fetal monitoring system with hardware-based data flow control,” in *Proc. IET 3rd Int. Conf. Adv. Med., Signal Inf. Process.*, 2006, pp. 1–4.

[55] J. Abonyi and F. Szeifert, “Supervised fuzzy clustering for the identification of fuzzy classifiers,” *Pattern Recognit. Lett.*, vol. 24, pp. 2195–2207, 2003.

[56] F. Klawonn and R. Kruse, “Derivation of fuzzy classification rules from multidimensional data,” *Adv. Intell. Data Anal.*, vol. 1, no. 1, pp. 90–94, 1995.

[57] P. P. Angelov and D. P. Filev, “An approach to online identification of Takagi-Sugeno fuzzy models,” *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 34, no. 1, pp. 484–498, Feb. 2004.

[58] A. Gomez-Skarmeta, M. Delgado, and M. Vila, “About the use of fuzzy clustering techniques for fuzzy model identification,” *Fuzzy Set Syst.*, vol. 106, no. 2, pp. 179–188, 1999.

[59] D. Nauck and R. Kruse, “A neuro-fuzzy method to learn fuzzy classification rules from data,” *Fuzzy Set Syst.*, vol. 89, no. 2, pp. 277–288, 1997.

[60] M. Setnes, “Supervised fuzzy clustering for rule extraction,” *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 4, pp. 416–424, Aug. 2000.

[61] Y. Zhang, H. Ishibuchi, and S. Wang, “Deep Takagi-Sugeno-Kang fuzzy classifier with shared linguistic fuzzy rules,” *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1535–1549, Jun. 2018.

Jacek M. Leski (M’98–SM’03) was born in Gliwice, Poland and received the M.Sc., Ph.D., and D.Sc. degrees in electronics from the Silesian University of Technology, Gliwice, Poland, in 1987, 1989, 1995, respectively.

He is a Full Professor of Biomedical Information Processing with the Silesian University of Technology, Gliwice, Poland, and the Head of its Division of Biomedical Electronics. He is also a Professor with the Łukasiewicz Research Network–Institute of Medical Technology and Equipment, Zabrze, Poland. He has authored and coauthored five monographs, 14 chapters in monographs, and more than 220 international journal and conference papers. His research interests include digital processing of biomedical signals, fuzzy and neuro-fuzzy modeling, pattern recognition, and learning theory.

Prof. Leski is a member of the Polish Society of Theoretical and Applied Electrotechnics. In 2017–2020, he was a member of the Scientific Policy Committee with the Polish Minister of Science and Higher Education.

Robert Czabański was born in Tychy, Poland. He received the M.Sc. and Ph.D. degrees in electronics, in 1997 and 2003, respectively, and the D.Sc. degree in biocybernetics and biomedical engineering, in 2018.

He is an Associate Professor with the Division of Biomedical Electronics of Silesian University of Technology, Gliwice, Poland. His research interests include fuzzy and neuro-fuzzy modeling, learning theory, and biomedical signal processing.

Dr. Czabanski is a member of the Polish Society of Theoretical and Applied Electrotechnics.

Michal Jezewski was born in Zabrze, Poland. He received the M.Sc. degree in computer science and the Ph.D. degree in electronics from the Silesian University of Technology, Gliwice, Poland, in 2006 and 2011, respectively.

He is an Assistant Professor with the Division of Biomedical Electronics of Silesian University of Technology, Gliwice, Poland. His research interests include biomedical signal processing, computational intelligence methods with emphasis on fuzzy clustering and fuzzy classifiers.

Dr. Jezewski is a member of the Polish Society of Theoretical and Applied Electrotechnics.

Janusz Jezewski (M’93–SM’06) was born in Zabrze, Poland. He received the M.Sc. degree in electronics from the Silesian University of Technology, Gliwice, Poland, Ph.D. degree in biological sciences from the University of Medical Sciences, Poznan, Poland, and D.Sc. degree in biocybernetics and biomedical engineering from the Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences, Warsaw, Poland.

He is an Associate Professor with the Łukasiewicz Research Network–Institute of Medical Technology and Equipment in Zabrze, Poland, where he is also a Director for science. He has authored and coauthored more than 300 international journal and conference papers. His research interests include biomedical instrumentation, digital signal processing and application of computational intelligence in medical cyber-physical systems.