Int. J. Appl. Math. Comput. Sci., 2021, Vol. 31, No. 2, 303–319
DOI: 10.34768/amcs-2021-0021

AN OUTLIER–ROBUST NEURO–FUZZY SYSTEM FOR CLASSIFICATION AND REGRESSION

KRZYSZTOF SIMINSKI [a]

[a] Department of Algorithmics and Software
Silesian University of Technology
ul. Akademicka 16, 44-100 Gliwice, Poland
email: krzysztof.siminski@polsl.pl

Real life data often suffer from non-informative objects—outliers. These are objects that are not typical in a dataset and can significantly decline the efficacy of fuzzy models. In the paper we analyse neuro-fuzzy systems robust to outliers in classification and regression tasks. We use the fuzzy c-ordered means (FCOM) clustering algorithm for scatter domain partition to identify premises of fuzzy rules. The clustering algorithm elaborates typicality of each object. Data items with low typicalities are removed from further analysis. The paper is accompanied by experiments that show the efficacy of our modified neuro-fuzzy system to identify fuzzy models robust to high ratios of outliers.

Keywords: outliers, neuro-fuzzy systems, clustering, classification, regression.

1. Introduction

A crucial feature of machine learning techniques is the ability to elaborate answers to questions unseen before. In order to answer first-time questions, an artificial intelligence method creates an internal representation of knowledge. Train data are used to create an internal representation of knowledge that it used to formulate answers to unseen (test) cases.

There are many artificial intelligence techniques. One of them are neuro-fuzzy systems. They elaborate a very convenient representation of knowledge: a set of fuzzy rules. Elaborated rules have intelligible form—they can be easily read by humans. A reverse approach is also possible: rules formulated by humans can be easily incorporated into a fuzzy rule base (Siminski, 2014). Rules proposed by humans in linguistic terms can be translated from the "human" to the "computer" language with application of fuzzy terms (Leski and Kotas, 2018).

Fuzzy set theory has twofold effect in neuro-fuzzy systems: (i) it transforms human terms into computer entities and (ii) it can better handle intrinsic imprecision of data (Sholla *et al*., 2020; Grzegorzewski *et al*., 2020; Piegat and Dobryakova, 2020). Lotfi Zadeh claims that the more complicated the model, the more suitable the fuzzy approach (Zadeh, 1973). Interpretability of a fuzzy model (a set of fuzzy rules) is an important feature of neuro-fuzzy systems extensively investigated recently (Riid, 2002; Cpałka *et al*., 2014; Słowik *et al*., 2020; Leski, 2015; Alcalá *et al*., 2006; Alonso and Magdalena, 2011; Evsukoff *et al*., 2009; Bartczuk *et al*., 2016; Otte, 2013).

Creation of a fuzzy rule base is commonly run in two steps: (i) identification of the model structure (the number of rules, attributes used, etc.) and (ii) identification of parameter values. The first step has a significant impact on the quality of an elaborated model. There are three common techniques: grid partition, scatter partition, and hierarchical partition of the input domain. Grid partition is the oldest approach and has many drawbacks, of which the curse of dimensionality is the most severe one (Matthews *et al*., 2013). This technique was used in the first neuro-fuzzy system ANFIS (Jang, 1993). Scatter partition is the most popular approach (Siminski, 2015; 2017c). It is based on the clustering of the input domain. Its main drawback is the necessity of providing a number of rules. The third approach (hierarchical partition) avoids drawbacks and takes advantages of both techniques above, but suffers from longer computation times (Jakubek and Keuth, 2006; Siminski, 2008; 2009; 2010).

An *outlier* is a commonly used term for a non-informative object that is not typical (with very

uncommon values of attributes). Outliers are sometimes called *outstanding data*. For informative data (typical) we use term *inlier* in this paper. Many techniques have been proposed for the clustering of data with outliers (Jiang and Yin, 2019; Jiang *et al*., 2018). The main approaches are preprocessing (e.g., 2σ-rule, 3σ-rule (Lehmann, 2013)), modification of the objective function in clustering (e.g., possibilistic clustering (Krishnapuram and Keller, 1993)), application of special metrics and quasi-norms (Hathaway *et al*., 2000), a special cluster for outliers (Dave and Krishnapuram, 1997), repulsive clusters (Timm *et al*., 2004), subtractive clustering (Yang *et al*., 2009), kernel density clustering (Latecki *et al*., 2007; Tang and He, 2017; Geng *et al*., 2018), manifold learning (Olson *et al*., 2018), the data ordering technique (Yager, 1988) and typicalities (Leski and Kotas, 2015; Leski, 2014; Siminski, 2020).

Identification of fuzzy rules is a crucial part in elaboration of a fuzzy model (a fuzzy rule base), and clustering is the most common technique of extraction of rule premises. This is why we focus on application of a clustering algorithm robust to outliers to construct a neuro-fuzzy system that can handle a wide range of outlier ratios in data sets.

2. Fuzzy c-ordered means clustering algorithm

The fuzzy c-ordered means (FCOM) clustering algorithm (Leski, 2014) addresses the vulnerability of the FCM algorithm (Dunn, 1973) to outliers and noise. The FCOM algorithm follows the Pickard iteration pattern used by FCM. It introduces two techniques to handle outliers: (i) modification of distance metrics and (ii) data ordering.

Modification of distance metrics. The Euclidean distance is a commonly used metric in clustering algorithms. Unfortunately, it is very vulnerable to outliers. The FCOM algorithm modifies the Euclidean distance with a loss function. The distance d between two items x_1 and x_2 is calculated with

$$d(x_1, x_2) = d_{\mathrm{Eu}}(x_1, x_2) \cdot h\left(|x_1 - x_2|\right), \qquad (1)$$

where d_{Eu} is a Euclidean distance and $h : \mathbb{R} \to \mathbb{R}^+ \cup \{0\}$ is a loss function defined in various ways (Table 1) (Leski, 2014). The idea of application of loss functions aims at reducing the influence of distant objects on the location of clusters. In the Euclidean metric, the distance is squared and outliers may have a significant but unjustified impact on the clustering procedure. Application of loss functions may reduce the impact of outliers (Kłopotek *et al*., 2020). Unfortunately, there is no theoretical basis on how to choose the optimal function. Thus, some preliminary experiments have to be run to find the best function. Some research suggests logarithmic and linear-logarithmic functions are a good choice (D'Urso and Leski, 2020; Siminski, 2017a).

Data ordering. The FCOM clustering algorithm applies an ordering technique. It orders all objects in each cluster by their distances from cluster centres. This is applied for each attribute separately. The closest object with respect to the d-th attribute is labelled with $k_d = 1$, the most distant with $k_d = X$ (where X is the number of objects). The label k is used to elaborate typicality of the objects in question with regard to the d-th attributes and the c-th clusters. We use the sigmoidally ordered weighted averaging function (SOWA) (Leski, 2014) of ordering number k,

$$\beta(k) = \frac{1}{1 + \exp\left(\frac{2.944}{aX}\left(k - cX\right)\right)}. \qquad (8)$$

The k-th object (where $k = cX$) has typicality 0.5. The value 2.944 in (8) is chosen so that for $a \cdot 100\%$ of objects the function has values greater than 0.95, and for $a \cdot 100\%$ the values are less than 0.05. The steepness of the function depends on the parameter a. The function is presented in Fig. 1 for $X = 100$ and $a = 0.2$, $c = 0.5$ (Siminski, 2017a). It is possible to use other functions, for instance, the piecewise linear OWA (PLOWA) function (Siminski, 2017a).

Typicality β_{cid} of the d-th attribute (descriptor) of the i-th object in the c-th cluster is elaborated for each attribute separately. We use here the 'a chain is only as strong as its weakest link' approach: typicality β_{ci} of the i-th object in the c-th cluster with regard to all attributes is no greater than the lowest typicality elaborated for any attribute. We model this approach with a t-norm ($\star$):

$$\beta_{ci} = \beta_{ci1} \star \beta_{ci2} \star \ldots \star \beta_{ciD}. \qquad (9)$$

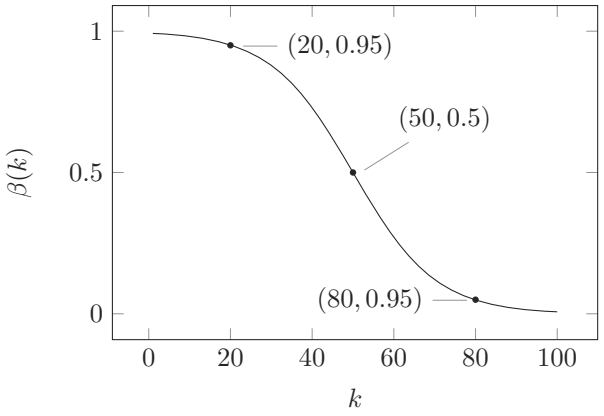

Fig. 1. SOWA weighting function (for $X = 100$ with $a = 0.2$ and $c = 0.5$. For $a \cdot 100\%$ (here: 20%) of data the values of the SOWA function are greater (resp. less) than 0.95 (resp. 0.05).

Table 1. Examples of loss functions.

Name of loss function	Formula
absolute (linear)	$h(x) = \begin{cases} 0, & x = 0 \\ \lvert x\rvert^{-1}, & x \neq 0 \end{cases}$ (2)
Huber (with parameter $\delta > 0$)	$h(x) = \begin{cases} \delta^{-2}, & x = 0 \\ (\delta\lvert x\rvert)^{-1}, & x \neq 0 \end{cases}$ (3)
sigmoidal (with parameters $\alpha, \beta > 0$)	$h(x) = \begin{cases} 0, & x = 0 \\ x^{-2}\left[1 + e^{(-\alpha(\lvert x\rvert - \beta)}\right]^{-1}, & x \neq 0 \end{cases}$ (4)
sigmoidal-linear (with parameters $\alpha, \beta > 0$)	$h(x) = \begin{cases} 0, & x = 0 \\ \lvert x\rvert^{-1}\left[1 + e^{-\alpha(\lvert x\rvert - \beta)}\right]^{-1}, & x \neq 0 \end{cases}$ (5)
logarithmic	$h(e) = \begin{cases} 0, & x = 0 \\ \frac{\log(1+x^2)}{x^2}, & x \neq 0 \end{cases}$ (6)
logarithmic-linear	$h(e) = \begin{cases} 0, & x = 0 \\ \frac{\log(1+x^2)}{\lvert x\rvert}, & x \neq 0 \end{cases}$ (7)

An object may be very typical for some cluster and atypical for others. This is why for the global typicality t_i of the i-th object we use an s-norm ($\diamond$) operator,

$$t_i = \beta_{1i} \diamond \beta_{2i} \diamond \ldots \diamond \beta_{Ci}. \tag{10}$$

In our experiments, we use the product t-norm and the maximum ($\vee$) s-norm. Thus, the global typicality can be expressed as

$$t_i = \bigvee_{c=1}^{C} \prod_{d=1}^{D} \beta_{cid}, \tag{11}$$

where C is the number of clusters and D is the number of attributes.

3. Neuro-fuzzy systems

In our experiments, we use TSK (Takagi and Sugeno, 1985; Sugeno and Kang, 1988) and ANNBFIS (Czogała and Łęski, 2000) neuro-fuzzy systems. Here we only highlight the main features of the systems. For details, please see the references mentioned above.

3.1. Architecture of the system. The neuro-fuzzy systems we use in our research are composed of two crucial components: the fuzzy rule base and the fuzzy inference engine. They take a vector of values (attributes) $\mathbf{x} = [x_1, x_2, \ldots, x_D]^{\mathrm{T}} \in \mathbb{R}^D$ as an input and elaborate one value $y \in \mathbb{R}$ as an output. Each rule l in a fuzzy rule base $\mathbb{L}$ is a fuzzy IF-THEN rule:

$$l : \text{IF } \mathbf{x} \text{ is } \mathfrak{a} \text{ THEN } y \text{ is } \mathfrak{b}, \tag{12}$$

where $\mathfrak{a}$ and $\mathfrak{b}$ are linguistic descriptions of inputs and output, respectively.

In our research we use two architectures of neuro-fuzzy systems: TSK and ANNBFIS. They have the same form of rule premises, but differ in the form of consequences and evaluation of rule values.

3.1.1. Premises of fuzzy rules. The premise of the l-th fuzzy rule is modelled with Gaussian fuzzy sets $\mathbb{A}_d$ in each dimension d. For the d-th dimension (attribute), a fuzzy set is defined with membership function u:

$$u_{\mathbb{A}_d}(x_d) = \exp\left(-\frac{(x_d - v_{ld})^2}{2s_{ld}^2}\right), \tag{13}$$

where v_{ld} is the core value for the d-th attribute and s_{ld} is the fuzziness of the attribute. We use the Gaussian membership function because it is differentiable in its whole domain. The memberships of all attributes (descriptors) are aggregated in order to elaborate the membership $u_{\mathbb{A}}$ of an object to the premise of the rule. A T-norm $\star$ is used as an aggregation operator (we use the product T-norm):

$$u_{\mathbb{A}} = u_{\mathbb{A}_1} \star u_{\mathbb{A}_2} \star \ldots \star u_{\mathbb{A}_D} = \prod_{d=1}^{D} u_{\mathbb{A}_d}. \tag{14}$$

The next step is a cooperation of ruzzy rules. The formulae (13) and (14) yield the activation (firing strength) F of the premise of l-th rule for object $\mathbf{x}$:

$$F_l(\mathbf{x}) = u_{l\mathbb{A}}(\mathbf{x}) = \prod_{d=1}^{D} \exp\left[-\frac{(x_d - v_{ld})^2}{2s_{ld}^2}\right], \tag{15}$$

which is a real number for any $\mathbf{x}$: $F(\mathbf{x}) \in (0, 1]$.

3.1.2. Consequences of fuzzy rules in the TSK system. The term $\mathfrak{b}$, in the formula (12), describing the l-th rule's consequence is represented by a fuzzy singleton. The localisation y_l of the singleton is determined by a linear combination of input attribute values:

$$\begin{aligned} y_l &= \mathbf{p}_l^{\mathrm{T}} \cdot \left[1, \mathbf{x}^{\mathrm{T}}\right]^{\mathrm{T}} \\ &= [p_{l0}, p_{l1}, \ldots, p_{lD}] \cdot \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_D \end{bmatrix} \\ &= \sum_{d=0}^{D} p_{ld} x_d, \end{aligned} \tag{16}$$

where $x_0 = 1$ and p_i's are linear coefficients. The height of the singleton is the firing strength $F_l(\mathbf{x})$ of the premise. The singletons of all rules are aggregated into the final answer of a TSK system with

$$y\left(\mathbf{x}\right) = \frac{\sum_{l=1}^{L} F_l\left(\mathbf{x}\right) y_l(\mathbf{x})}{\sum_{l=1}^{L} F_l\left(\mathbf{x}\right)}. \tag{17}$$

3.1.3. Consequences of fuzzy rules in the ANNBFIS system. The term $\mathfrak{b}$, in the formula (12), describing the l-th rule's consequence is a normal isosceles triangle fuzzy set $\mathbb{B}_l$ with the base width w_l (as in the Mamdami–Assilan architecture (Mamdani and Assilian, 1975)). The localisation y_l of the core of the triangle fuzzy set is determined by a linear combination (Eqn. (16)) of input attribute values (as for a singleton in the Takagi–Sugeno–Kang architecture).

In the ANNBFIS architecture the IF-THEN rule is a true logical fuzzy implication. The membership function $u_{l\mathbb{B}'}$ of the l-th rule is a fuzzy value of the fuzzy implication:

$$u_{\mathbb{B}'_l}\left(\mathbf{x}\right) = u_{l\mathbb{A}}\left(\mathbf{x}\right) \rightsquigarrow u_{\mathbb{B}_l}\left(\mathbf{x}\right), \tag{18}$$

where $u_{l\mathbb{A}}\left(\mathbf{x}\right)$ is a membership of the $\mathbf{x}$ tuple to the fuzzy set $\mathbb{A}$ in the l-th rule (Eqn. (15)), $u_{\mathbb{B}_l}$ is the membership function in the consequence, and the squiggle arrow ($\rightsquigarrow$) stands for a fuzzy implication. The shape of the fuzzy set $\mathbb{B}'$ depends on the fuzzy implication used (Czogała and Łęski, 2000). In the system, the Reichenbach implication (Reichenbach, 1935) is used:

$$p \rightsquigarrow q = 1 - p + pq. \tag{19}$$

It is possible to use other fuzzy implications such as the Łukasiewicz, Kleene–Dienes, Rescher, Goguen, Gödel, or Zadeh ones. There is no theoretical research on the choice of the implication. Multiple experiments point out the Reichenbach implication is effective in neuro-fuzzy systems with logical interpretation of fuzzy rules (Siminski, 2017b).

The answers $\mathbb{B}'_l$ of all L rules are then aggregated into one fuzzy answer of the system:

$$\mathbb{B}'\left(\mathbf{x}\right) = \bigoplus_{l=1}^{L} \mathbb{B}'_l\left(\mathbf{x}\right), \tag{20}$$

where $\bigoplus$ stands for the aggregation operator. The fuzzy answer $\mathbb{B}'$ is defuzzified in a crisp (non-fuzzy) value y_0 with the MICOG method (Czogała and Łęski, 2000). This approach aggregates only the informative part of implication sets. In general terms the defuzzification procedure may be quite expensive, but it has been proved (Czogała and Łęski, 2000) that combining aggregation and defuzzification in the MICOG method can be expressed with

$$y\left(\mathbf{x}\right) = \frac{\sum_{l=1}^{L} g_l\left(\mathbf{x}\right) y_l(\mathbf{x})}{\sum_{l=1}^{L} g_l\left(\mathbf{x}\right)}. \tag{21}$$

The functions g depends on the fuzzy implication; in the system, the Reichenbach one is used, so for the l-th rule function g is

$$g_l\left(\mathbf{x}\right) = \frac{1}{2} w_l u_{l\mathbb{A}}\left(\mathbf{x}\right). \tag{22}$$

For function g for other applicable implications, please see the original work introducing the ANNBFIS system, (Czogała and Łęski, 2000) (with some inaccuracies discussed by Nowicki (2006)).

3.2. Forming the fuzzy model. The fuzzy model (the fuzzy rule base) is created in three steps:

(i) partition of the input domain with the fuzzy c-means clustering (Dunn, 1973) and fuzzy c-ordered means clustering (Leski, 2014) algorithms,

(ii) extraction of premises of rules from clusters (Section 3.2.2) (de Souza, 2020; Dovžan and Škrjanc, 2011; Siminski, 2012),

(iii) tuning of rules (Section 3.2.3) (Seresht and Fayek, 2020; Škrjanc *et al.*, 2019; Siminski, 2016; Harifi *et al.*, 2020).

Organisation of a system with the FCOM clustering algorithm is presented in Fig. 2. The algorithm has a twofold task: (i) identification of rule premises and (ii) identification of outliers. Outliers are removed from the training data set and the model is further tuned with inliers only. This makes the tuning procedure faster and more reliable because outliers do not interfere and disturb the tuning of parameters.

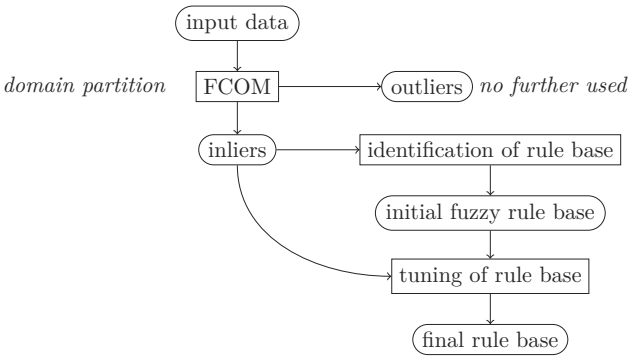

Fig. 2. System organisation.

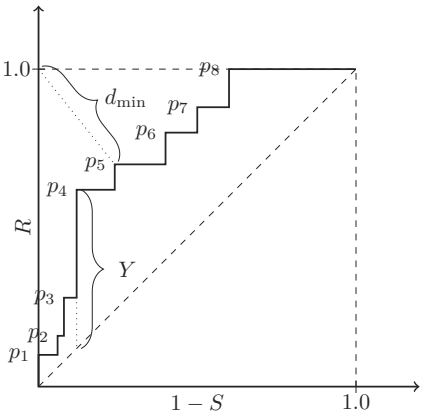

Fig. 3. ROC curve for a classifier with eight values of threshold. Each value results in one point p_i on the curve. R stands for recall, S for specificity, $p_1, \ldots, p_8$ for thresholds, Y for the Youden index, $d_{\min}$ for the distance of the closest point of the ROC curve to the $(0, 1)$ point. With the Youden index, p_4 is chosen, with the minimal distance $d_{\min}$ corresponding to p_5.

3.2.1. Partition of the input domain. Because premises of fuzzy rules split the domain into regions, we use the reverse approach: we split the input domain in order to identify rule premises. For partition of the input domain we use the scatter approach. The domain is partitioned with two clustering algorithms FCM (Dunn, 1973) and FCOM (Leski, 2014). The former is a commonly used clustering algorithm—we do not describe it in the paper. The latter is a clustering algorithm that applies two techniques to handle outliers: the ordering of data and modification of distance metrics.

In an FCM cluster, centres are determined with Langrange multiplies and fortunately the localisation of cluster centres may be expressed with a closed-form formula. Unfortunately, in FCOM, cluster centres are determined in an iterative way. Objects with low typicalities have lower influence on localisation the cluster centres.

The FCOM is used not only to partition the input domain of the task, but also to elaborate typicalities of objects. Items with typicalities below the typicality cut-off are removed from the train data set and are not used for tuning the neuro-fuzzy system's parameters.

3.2.2. Extraction of rules from clusters. The FCOM clustering procedure estimates the locations of cluster centres (gathered in the $\mathbf{U} = \{u_{ij}\}$ matrix) and typicality of each object in the data set ($\mathbf{t} = [t_1, \ldots, t_X]$). The number of rules equals the number of clusters: $L = C$. The membership matrix $\mathbf{U}$ is used to calculate the centres of a cluster that are the cores of premises, and fuzzification of descriptors in premises.

3.2.3. Tuning of rule parameters. In neuro-fuzzy systems, the parameters of the model are tuned to better fit the data. The premises are built with Gaussian fuzzy sets. Because a Gaussian function is differentiable for all arguments, we can apply a gradient optimisation method. We use this technique for optimisation of parameters of premises and support width w in the ANNBFIS consequences. Linear coefficients p_i (Eqn. (16)) for the calculation of the localisation of the cores of fuzzy sets in the consequences of rules are calculated precisely with linear regression.

3.3. Class separation threshold. The systems used in the experiments produce continuous answer that has to be mapped into values representing positive and negative classes. We have to decide which values should be labelled with a positive and which with a negative class. We use the ROC curve to find an optimal threshold. We sort all answers produced for a train set and analyse all possible thresholds. For each threshold we calculate recall R (25) and specificity S (26), and plot the ROC curve. We use three methods to find the optimal threshold:

- the mean value of both positive and negative classes (the only method that does not need the ROC curve); e.g., if the negative class is 0 and the positive class is 1, then the threshold is 0.5;
- the Youden index Y (Youden, 1950): we analyse all possible thresholds using the Youdex index

$$Y = R + S - 1 \tag{23}$$

(where R stands for recall, Eqn. (25), and S for specificity, Eqn. (26)) and choose the threshold with the maximal Youden index Y; the Youden index has a simple geometrical interpretation (Fig. 3).

- the minimal distance $d_{\min}$ criterion: we analyse all possible thresholds p_i and choose the one

corresponding to the point closest to $(0, 1)$ on the ROC curve (Fig. 3).

Objects for which classifiers return a value above the threshold are labelled with a positive class, otherwise with a negative one.

4. Experiments

4.1. Datasets. All data sets are prepared in the same way. The data are not normalised. The set is partitioned at random into train and test data sets of the same size.

A set of outliers is added to the train test in each data set. The ratio of the added outliers is 30% of the original items in the train set. In each outlier all non-decisive attributes are generated from the Gaussian distribution $N_a(m, \sigma)$ with the average and standard deviation defined separately for each attribute. In each outlier one attribute has an extreme value. It is generated with Gaussian distribution $N_o(10m, 0.1\sigma)$. In outliers the decisive attribute is generated with the same probability as in the original train set. Thus the distribution of classes is very close to the original distribution in the training set.

We add outliers only to the train data set, because we do not want systems to learn the outliers, but to ignore them. We do not add outliers to test sets, to verify whether a system manages to detect and ignore outliers.

Experiments were run for all data sets with multiple values of parameters:

- (for classification) three types of threshold calculation: the mean, minimal distance, and Youden criterion (cf. Section 3.3);
- the number of rules in neuro-fuzzy systems: 3, 4, 5, ..., 12;
- typicality cut-off: 10^{-12}, 10^{-11}, 10^{-10}, ..., 10^{-1} (cf. Section 3.2.1);
- neuro-fuzzy systems: TSK, ANNBFIS, FCOM-TSK, FCOM-ANNBFIS;
- the experiment for each set of parameters was repeated 13 times.

The total number of experiments is 70980 for classification and 82810 for regression. We use a fast free C++ implementations from the Neuro-Fuzzy Library of the TSK and ANNBFIS neuro-fuzzy systems (Siminski, 2019) for our modifications and experiments.

In all experiments we use the logarithmic-linear loss function. We have chosen the logarithmic-linear function based on preliminary experiments. For brevity, we do not present the comparison of results for different loss functions.

We run the experiments both for classification and regression. All data sets for classification can be downloaded from the UCI data repository (Frank and Asuncion, 2019). Essential statistics of data sets used for classification are gathered in Table 2 and for regression—in Table 3. The tables present the distribution of classes in training data sets. The distribution in test sets is very similar. Data sets for the regression tasks: 'power', 'beijing', and 'carbon' can be downloaded from the UCI repository (Frank and Asuncion, 2019). The data set 'synthetic' is first defined in this paper.

Synthetic data set ('synthetic'). The 'synthetic' data set is a two-dimensional surface presented in Fig. 4. It has been created with four fuzzy rules:

Rule 1: IF x_1 is low AND x_2 is low THEN y is small,
Rule 2: IF x_1 is low AND x_2 is high THEN y is large,
Rule 3: IF x_1 is high AND x_2 is low THEN y is large,
Rule 4: IF x_1 is high AND x_2 is high THEN y is small.

Linguistic terms are modelled with Gaussian fuzzy sets.

4.2. Measures. We use different quality measures of classification and regression.

4.2.1. Classification. For the classification task we use the confusion table (Fig. 4) to analyse results.

We use accuracy defined as (we use notation defined in Fig. 4, the symbol $[\rightarrow]$ stands for number of all items)

$$A = \frac{[p \rightarrow p] + [n \rightarrow n]}{[p \rightarrow p] + [n \rightarrow p] + [n \rightarrow n] + [p \rightarrow n]} = \frac{[p \rightarrow p] + [n \rightarrow n]}{[\rightarrow]}, \tag{24}$$

Recall (sensitivity, hit rate, probability of detection, power) is defined as

$$R = \frac{[p \rightarrow p]}{[p \rightarrow p] + [p \rightarrow n]} = \frac{[p \rightarrow p]}{[p \rightarrow]} \tag{25}$$

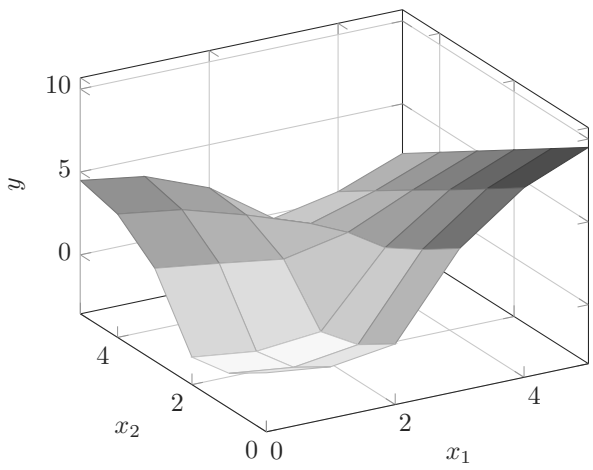

Fig. 4. Surface defined by the 'synthetic' data set.

Table 2. Essential statistics of data sets for classification. The table presents the distributions of classes in training data sets (without outliers). Test data sets have very similar distribution of classes.

Dataset	Number of attributes	tuples	Distribution of classes negative		positive		References
'banknote'	5	1372	378	(55.1%)	308	(44.9%)	Frank and Asuncion, 2019
'wilt'	6	4889	487	(97.4%)	13	(2.6%)	Johnson *et al.*, 2013
'vertebral'	6	310	52	(33.5%)	103	(66.5%)	Rocha Neto and Barreto, 2009
'htru'	9	2000	919	(91.9%)	81	(8.1%)	Keith *et al.*, 2010; Lyon *et al.*, 2016
'haberman'	3	306	116	(75.8%)	37	(24.1%)	Haberman, 1976
'parkinsons'	23	197	24	(24.0%)	76	(76.0%)	Frank and Asuncion, 2019
'blood'	5	748	277	(74.1%)	97	(25.9%)	Yeh *et al.*, 2009

Table 3. Essential statistics of datasets for regression.

Dataset	attributes	train tuples original	train tuples with outliers	test tuples	References
'synthetic'	2	400	520	400	This paper
'CO_2'	12	2653	3448	2654	Sikora and Krzykawski, 2005
'carbon'	5	5360	6968	5360	Acı and Avcı, 2016; 2017
'power'	4	4784	6219	4784	Tüfekci, 2014; Kaya *et al.*, 2012
'beijing'	4	1000	1300	1000	Liang *et al.*, 2015

Table 4. Confusion table for binary classification.

		Predicted class positive $[\rightarrow p]$	Predicted class negative $[\rightarrow n]$
Original class	positive $[p \rightarrow]$	$[p \rightarrow p]$ true positive	$[p \rightarrow n]$ false negative
	negative $[n \rightarrow]$	$[n \rightarrow p]$ false positive	$[n \rightarrow n]$ true negative

and specificity (selectivity or true negative rate) as

$$S = \frac{[n \rightarrow n]}{[n \rightarrow n] + [n \rightarrow p]} = \frac{[n \rightarrow n]}{[n \rightarrow]}. \qquad (26)$$

4.2.2. Regression. For the regression task we use the root mean square error (RMSE) defined as

$$E_{\mathrm{RMSE}}(\mathbb{X}) = \sqrt{\frac{1}{X}\sum_{i=1}^{X}[y(\mathbf{x}_i) - \hat{y}(\mathbf{x}_i)]^2}, \qquad (27)$$

where $\mathbb{X}$ is a data set (for which the RMSE is calculated) with X objects. Given x_i, the actual value of the response is denoted as $y(\mathbf{x}_i)$, whereas the corresponding calculated (predicted) value is $\hat{y}(\mathbf{x}_i)$.

4.3. Results of experiments.

4.3.1. Classification. In the presentation of results we show only accuracy. Most of the data sets are unbalanced and such a presentation may seem faulty. However, our approach produces accuracy $A = 1.0$ that means a perfect classification with recall and sensitivity also equals 1.0. The medians of accuracies produced by the neuro-fuzzy systems with our approach are presented in Table 5. The default accuracy is the frequency of the majority class in a data set.

It can be easily seen that the technique most common in practice for threshold determination (the mean value of classes) is the worst approach. It is better to use the minimal distance criterion or the Youden index. For all tested data sets the ANNBFIS neuro-fuzzy system with the FCOM clustering algorithm for fuzzy model identification reaches a 1.000 accuracy. This means that all test objects have been correctly classified. The simpler TSK neuro-fuzzy system with FCOM reaches the same accuracy for all datasets except for 'banknote'. It is also worth mentioning that our modification results in more compact fuzzy models (with fewer rules).

Figure 5 presents classification accuracy of the 'parkinsons' data set for the ANNBFIS neuro-fuzzy system with FCM and FCOM (with various typicality cut-offs). We can notice that the ANNBFIS system cannot reach the default accuracy with models with four and more rules.

Figure 6 presents, analogously, the accuracy for the 'vertebral' data set and the TSK neuro-fuzzy system.

Figures 7 and 8 present maximal, average, and minimal typicalities of inliers and outliers for the 'htru' and 'blood' data sets, respectively. The ratio between

Table 5. Accuracy of classification. *Italicised accuracy denotes values not greater than the default majority class frequency.* 'All' in the 'rules' columns stands for models with 3, 4, ..., 12 fuzzy rules.

Dataset (default accuracy)	Threshold type	TSK FCM accuracy	TSK FCM rules	TSK FCOM accuracy	TSK FCOM rules	ANNBFIS FCM accuracy	ANNBFIS FCM rules	ANNBFIS FCOM accuracy	ANNBFIS FCOM rules
'banknote'	mean	0.9905	7	0.9970	3	1.0000	11	0.989	3
(0.551)	Youden	0.9985	7	0.9970	3	1.0000	11	1.000	3
	min. dist.	0.9985	7	0.9948	3	1.0000	11	1.000	3
'wilt'	mean	*0.7180*	10	*0.6970*	11	*0.6730*	10	*0.780*	12
(0.974)	Youden	*0.764*0	12	1.0000	all	*0.6950*	11	1.000	all
	min. dist.	*0.7340*	12	1.0000	all	*0.6960*	9	1.000	all
'vertebral'	mean	0.8516	3	0.8451	8	0.8516	7	0.854	9
(0.665)	Youden	0.8516	4	1.0000	4	0.8419	9	1.000	all
	min. dist.	0.8516	4	1.0000	3	0.8258	5	1.000	all
'htru'	mean	0.9780	12	0.9270	10	0.9820	4	0.926	11
(0.919)	Youden	0.9730	12	1.0000	all	0.9660	5	1.000	all
	min. dist.	0.9720	12	1.0000	all	0.9530	12	1.000	all
'haberman'	mean	*0.7418*	10	*0.7582*	11	*0.7516*	9	*0.748*	7
(0.758)	Youden	*0.7582*	5	1.0000	all	0.7778	3	1.000	all
	min. dist.	*0.7582*	5	1.0000	all	0.7843	3	1.000	all
'parkinsons'	mean	*0.7253*	12	*0.4894*	5	*0.7211*	8	*0.415*	6
(0.760)	Youden	1.0000	4	1.0000	all	*0.6842*	7	1.000	all
	min. dist.	1.0000	4	1.0000	all	*0.8263*	3	1.000	all
'blood'	mean	0.7941	5	0.7834	all	0.8128	5	0.783	all
(0.741)	Youden	0.7540	6	1.0000	all	0.7459	5	1.000	all
	min. dist.	*0.7299*	10	1.0000	all	0.7459	5	1.000	all

average typicalities of inliers and outliers is approximately 10^{10} for the 'htru' dataset and 10^5 for 'blood'. We do not present similar plots for other data sets because they are very similar. In Table 6 we gather minimal, average, and maximal typicalities for models with five rules produced by our system for all data sets. The difference between typicalities elaborated for inliers and outliers is significant. This is why we use this technique to detect outliers. The smallest difference is obtained for the 'haberman' dataset.

4.3.2. Regression. The results of experiments are presented in Figs. 10, 12, 14, 16, 18, 20, 22, 24, 26 and 28.

The results for the 'carbon' data set present a characteristic slope of the error surface. The threshold cut-off value 10^{-6} splits the results into two parts: in one the outliers are correctly identified and removed from model elaboration, in the second part the outliers heavily distort the elaborated model. In experiments for both neuro-fuzzy systems (Figs. 10 and 12) the typicality approach outperforms the reference neuro-fuzzy systems.

The results for the 'synthetic' data set are presented in Figs. 14 and 16. The 3D plot presents the RMSE for a grid of a number of rules and typicality cut-offs. The 2D plot is less readable, but also presents the reference system's results (with the FCM clustering algorithm). For the 'synthetic' data set we can see that a typicality cut-off threshold is 10^{-5} for both the TSK and ANNBFIS systems. If the typicality cut-off is lower, the outliers start to influence the elaboration of a fuzzy model and deteriorate its quality. For this data set we can see a window for the best number of rules. If the number of rules is too low (three or four rules), the model is too weak. If the number of rules is too high (eight and more rules), the model overfits the data. For a typicality cut-off that removes outliers our modification of the NFS elaborates fuzzy models that outperform the reference NFSs.

Similar conclusions can be drawn for the 'power' and 'beijing' data sets. A typicality cut-off threshold can be found at 10^{-6} (Fig. 18). However, for the ANNBFIS based system the low number of rules leads to a poorer model and higher error ratios (Fig. 20). For ten and more rules the model overfits the data and loses the generalisation ability. For the 'beijing' data set the low number of rules is not sufficient and the elaborated model results in high error rates (Fig. 22).

This phenomenon is not clearly seen for the 'CO_2' data set. However, we can observe that neuro-fuzzy systems with typicalities can outperform an atypical neuro-fuzzy system and yield a lower error rate (RSME) for test data. This can be seen in Fig. 27, where the

Table 6. Examples of typicalities (minimal, average, and maximal) for inliers and outliers in models with 5 rules.

	Typicalities					
	inliers			outliers		
Dataset	min	avg	max	min	avg	max
'banknote'	$6.37 \cdot 10^{-19}$	$1.20 \cdot 10^{-1}$	$10.00 \cdot 10^{-1}$	$5.25 \cdot 10^{-17}$	$1.73 \cdot 10^{-7}$	$2.33 \cdot 10^{-6}$
'wilt'	$1.51 \cdot 10^{-24}$	$7.78 \cdot 10^{-2}$	$9.99 \cdot 10^{-1}$	$6.72 \cdot 10^{-23}$	$2.56 \cdot 10^{-8}$	$1.23 \cdot 10^{-6}$
'vertebral'	$1.33 \cdot 10^{-27}$	$6.16 \cdot 10^{-2}$	$9.97 \cdot 10^{-1}$	$2.39 \cdot 10^{-22}$	$5.07 \cdot 10^{-9}$	$2.05 \cdot 10^{-7}$
'haberman'	$2.68 \cdot 10^{-12}$	$2.76 \cdot 10^{-1}$	$10.00 \cdot 10^{-1}$	$5.10 \cdot 10^{-15}$	$2.30 \cdot 10^{-2}$	$10.00 \cdot 10^{-1}$
'parkinsons'	$2.25 \cdot 10^{-89}$	$2.18 \cdot 10^{-2}$	$9.79 \cdot 10^{-1}$	$1.71 \cdot 10^{-62}$	$5.14 \cdot 10^{-30}$	$1.49 \cdot 10^{-28}$
'blood'	$3.62 \cdot 10^{-22}$	$1.56 \cdot 10^{-1}$	$10.00 \cdot 10^{-1}$	$1.03 \cdot 10^{-16}$	$1.11 \cdot 10^{-7}$	$1.48 \cdot 10^{-6}$
'htru'	$1.79 \cdot 10^{-43}$	$7.28 \cdot 10^{-2}$	$9.98 \cdot 10^{-1}$	$7.07 \cdot 10^{-38}$	$1.01 \cdot 10^{-13}$	$2.85 \cdot 10^{-11}$

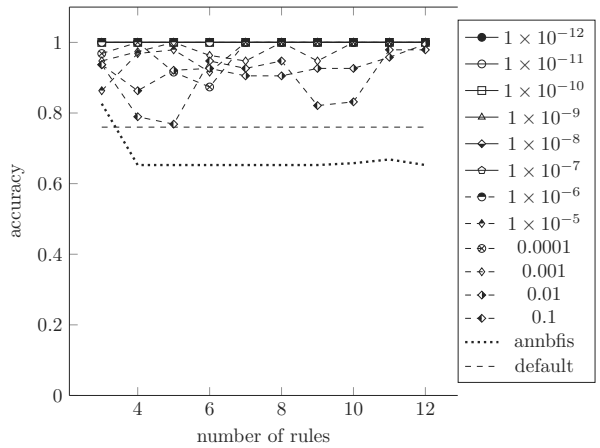

Fig. 5. Classification accuracy of the 'parkinsons' data set with the ANNBFIS system with FCOM and various typicality cut-offs ('annbfis' stands for the reference ANNBFIS system with the FCM clustering algorithm, 'default' for the majority class frequency in the data set).

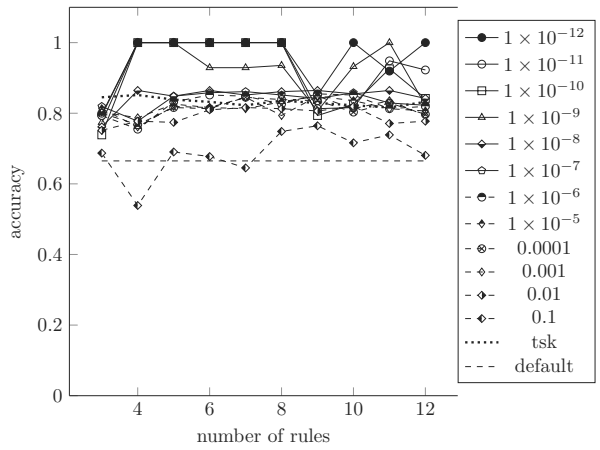

Fig. 6. Classification accuracy of the 'vertebral' data set with the TSK system with FCOM and various typicality cut-offs ('tsk' stands for the reference TSK system with the FCM clustering algorithm, 'default' for the majority class frequency in the data set).

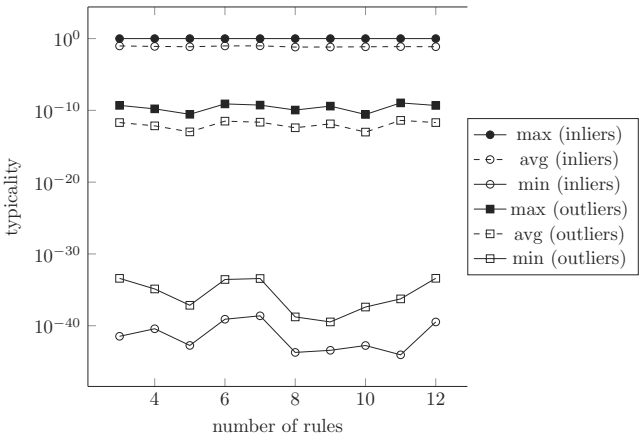

Fig. 7. Extreme and average values of typicalities of informative objects (inliers) and non-informative items (outliers) in the 'htru' dataset produced by the ANNBFIS neuro-fuzzy system with the FCOM clustering algorithm.

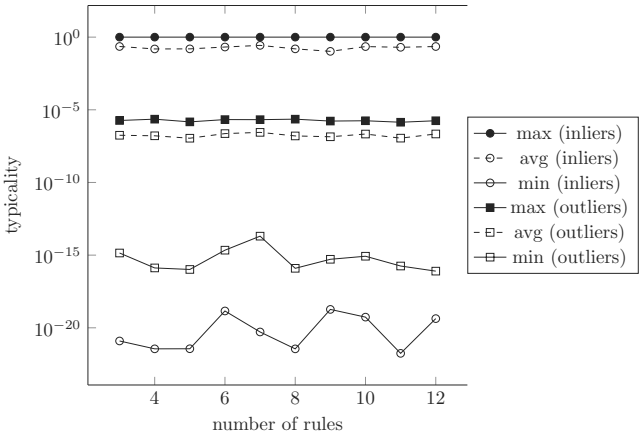

Fig. 8. Extreme and average values of typicalities of informative objects (inliers) and non-informative items (outliers) in the 'blood' data set produced by the ANNBFIS neuro-fuzzy system with the FCOM clustering algorithm.

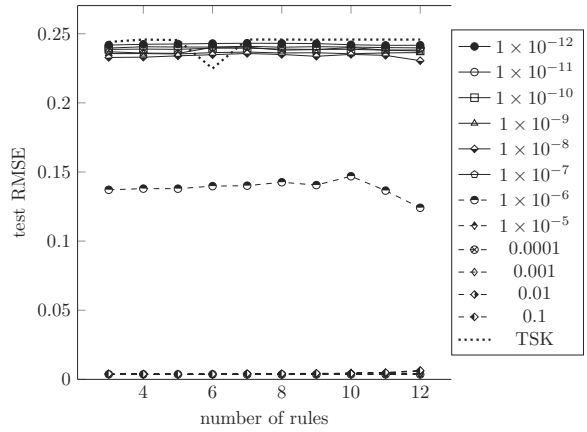

Fig. 9. Root mean square errors for the 'carbon' data set with the FCOM-TSK neuro-fuzzy system.

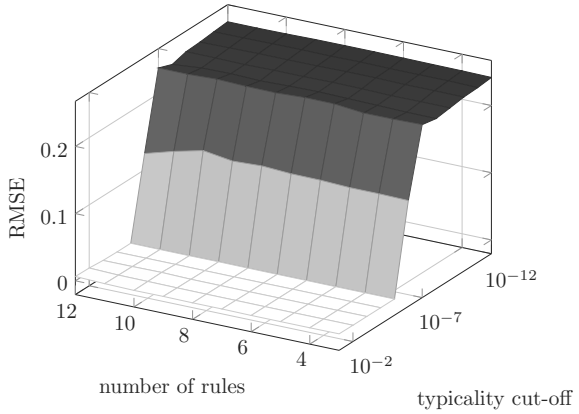

Fig. 10. Root mean square errors for the 'carbon' data set with the FCOM-TSK neuro-fuzzy system.

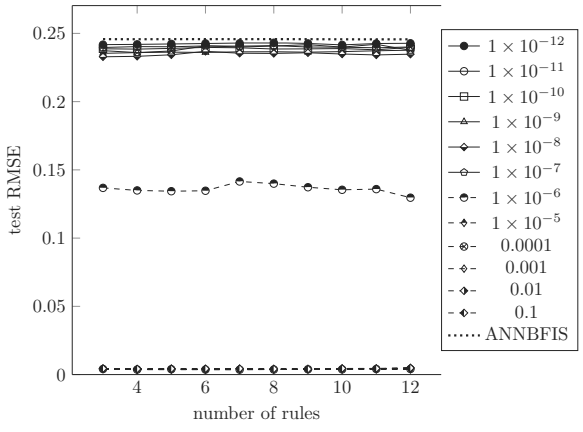

Fig. 11. Root mean square errors for the 'carbon' data set with the FCOM-ANNBFIS neuro-fuzzy system.

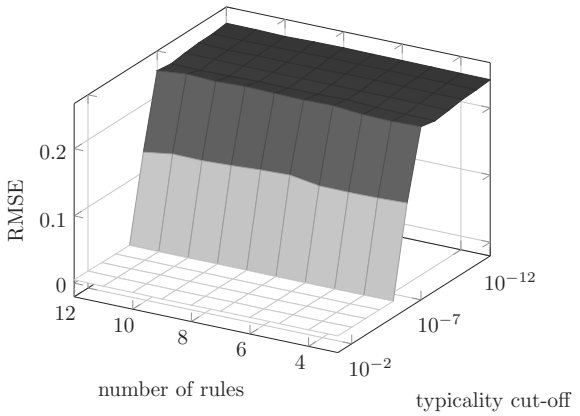

Fig. 12. Root mean square errors for the 'carbon' data set with the FCOM-ANNBFIS neuro-fuzzy system.

dotted line represents results elaborated by the ANNBFIS reference system.

4.4. Comparison with the 3σ method. The 3σ rule is a common technique to detect outliers. For each attribute in a dataset, the mean m and the standard deviation σ are determined. The values exceeding $m \pm 3\sigma$ are treated as outliers (Hekimoglu *et al.*, 2015; Lehmann, 2013). Some researchers use this approach to define outliers (Hekimoglu and Koch, 2000). This method also has some disadvantages, e.g., the mean value itself m is prone to outliers (Leys *et al.*, 2013). Figures 29–33 present a comparison of the proposed method with preprocessing with the 3σ heuristics to identify and remove outliers from a training data set that are not used in the identification and tuning of a fuzzy rule base of the TSK and ANNBFIS neuro-fuzzy systems. For lower ratios of outliers it is very hard to tell which method (ours or 3σ preprocessing) is more effective. For higher values of outliers our approach produced fuzzy models with lower errors in the 'beijing' (Fig. 29), 'carbon' (Fig. 31), 'CO_2' (Fig. 32), 'synthetic' (Fig. 33) data sets. For the 'carbon' data set the superiority of our method is not very significant. The 'test RSME' axis does not start with zero to better show the difference (Fig. 31). In the 'power' (Fig. 30) data set the most effective performance of our method is for medium outlier ratios.

The same methodology is the classification task. Table 7 presents a comparison of 3σ preprocessing and our method for the classification task. We omit ratios of added outliers 0.01, 0.02, 0.05, because both the approaches handle such ratios very well. In classification our approach can better handle high ratios of outliers than the 3σ preprocessing. We do not present results for $3\sigma+$ ANNFIS and FCOM-ANNBFIS neuro-fuzzy systems–they are similar to the results presented in Table 7 for the $3\sigma+$ TSK and FCOM-TSK techniques.

5. Conclusions

Identification of fuzzy rules is a crucial part in elaboration

Table 7. Classification accuracy produced by TSK neuro-fuzzy systems with 3σ preprocessing and TSK with FCOM premise identification. Both systems apply the Youden index (Eqn. (23)).

		Added outliers ratio				
Data set	Method	0.10	0.20	0.30	0.40	0.50
'banknote'	$3\sigma+$ tsk	1.0000	0.9985	0.9927	0.9388	0.7974
	fcom-tsk	1.0000	0.9985	0.9956	0.9636	0.9067
'wilt'	$3\sigma+$ tsk	1.0000	1.0000	1.0000	0.9660	0.9140
	fcom-tsk	1.0000	1.0000	1.0000	1.0000	0.9640
'vertebral'	$3\sigma+$ tsk	1.0000	1.0000	0.9790	0.9419	0.8903
	fcom-tsk	1.0000	1.0000	1.0000	1.0000	0.9806
'htru'	$3\sigma+$ tsk	1.0000	1.0000	1.0000	0.9710	0.9750
	fcom-tsk	1.0000	1.0000	1.0000	1.0000	0.9580
'haberman'	$3\sigma+$ tsk	1.0000	1.0000	1.0000	0.9412	0.9281
	fcom-tsk	1.0000	1.0000	1.0000	1.0000	1.0000
'parkinsons'	$3\sigma+$ tsk	1.0000	1.0000	1.0000	1.0000	0.9300
	fcom-tsk	1.0000	1.0000	1.0000	1.0000	0.8900
'blood'	$3\sigma+$ tsk	1.0000	1.0000	0.9840	0.9813	0.9893
	fcom-tsk	1.0000	1.0000	1.0000	1.0000	0.9920

of a fuzzy model (fuzzy rule base). Clustering is the most common technique of extraction of rule premises for neuro-fuzzy systems. In the paper we use a fuzzy c-ordered algorithm that applies modification of a distance measure and typicalities to detect outliers and reduce their impact on clustering. Application of this algorithm for fuzzy model identification makes neuro-fuzzy systems robust to outliers. FCOM successfully identifies the outliers and assigns them low typicalities. This makes it possible to exclude them and ignore in the fuzzy model identification process. Application of the method to both regression and classification tasks shows that this approach can successfully identify outliers and create a fuzzy model robust to high ratios of outliers.

Acknowledgment

This work was co-financed with a Silesian University of Technology grant for the maintenance and development of research potential.

References

Acı, M. and Avcı, M. (2016). Artificial neural network approach for atomic coordinate prediction of carbon nanotubes, *Applied Physics A* **122**(631).

Acı, M. and Avcı, M. (2017). Reducing simulation duration of carbon nanotube using support vector regression method, *Journal of the Faculty of Engineering and Architecture of Gazi University* **32**(3): 901–907.

Alcalá, R., Alcalá-Fdez, J., Casillas, J., Cordón, O. and Herrera, F. (2006). Hybrid learning models to get the interpretability-accuracy trade-off in fuzzy modeling, *Soft Computing* **10**(9): 717–734.

Alonso, J.M. and Magdalena, L. (2011). Special issue on interpretable fuzzy systems, *Information Sciences* **181**(20): 4331–4339.

Bartczuk, L., Przybyl, A. and Cpalka, K. (2016). A new approach to nonlinear modelling of dynamic systems based on fuzzy rules, *International Journal of Applied Mathematics and Computer Science* **26**(3): 603–621, DOI: 10.1515/amcs-2016-0042.

Cpałka, K., Łapa, K., Przybył, A. and Zalasiński, M. (2014). A new method for designing neuro-fuzzy systems for nonlinear modelling with interpretability aspects, *Neurocomputing* **135**: 203–217.

Czogała, E. and Łęski, J. (2000). *Fuzzy and Neuro-Fuzzy Intelligent Systems*, Physica-Verlag, Heidelberg/New York.

Dave, R. and Krishnapuram, R. (1997). Robust clustering methods: A unified view, *Fuzzy Systems, IEEE Transactions on* **5**(2): 270–293.

de Souza, P.V.C. (2020). Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature, *Applied Soft Computing* **92**: 106275.

Dovžan, D. and Škrjanc, I. (2011). Recursive fuzzy c-means clustering for recursive fuzzy identification of time-varying processes, *ISA Transactions* **50**(2): 159–169.

Dunn, J.C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters, *Journal Cybernetics* **3**(3): 32–57.

D'Urso, P. and Leski, J.M. (2020). Fuzzy clustering of fuzzy data based on robust loss functions and ordered weighted averaging, *Fuzzy Sets and Systems* **389**: 1–28.

Evsukoff, A.G., Galichet, S., de Lima, B.S. and Ebecken, N.F. (2009). Design of interpretable fuzzy rule-based classifiers using spectral analysis with structure and parameters optimization, *Fuzzy Sets and Systems* **160**(7): 857–881.

Frank, A. and Asuncion, A. (2019). UCI machine learning repository, `http://archive.ics.uci.edu/ml`.

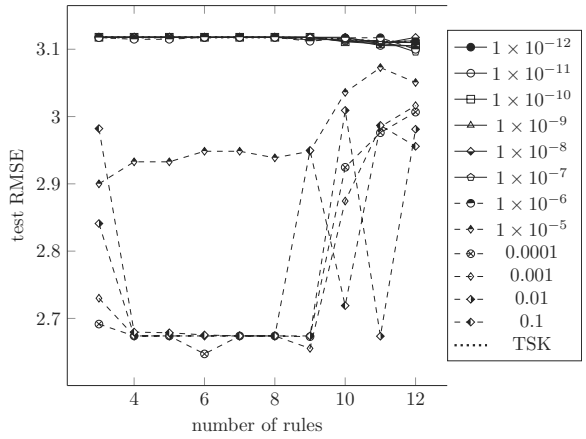

Fig. 13. Root mean square errors for the 'synthetic' data set with the FCOM-TSK neuro-fuzzy system.

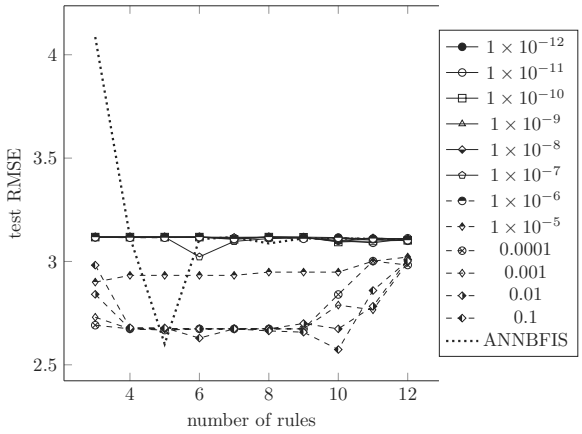

Fig. 15. Root mean square errors for the 'synthetic' data set with the FCOM-ANNBFIS neuro-fuzzy system.

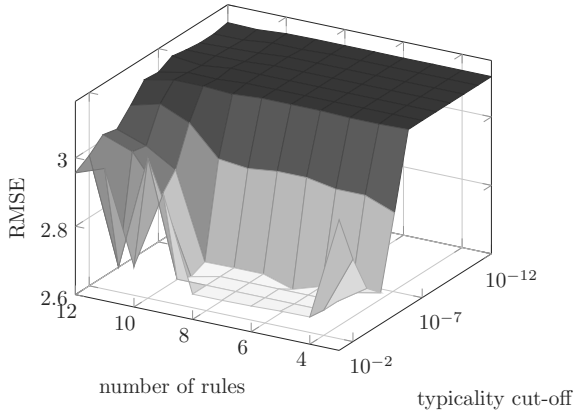

Fig. 14. Root mean square errors for the 'synthetic' data set with the FCOM-TSK neuro-fuzzy system.

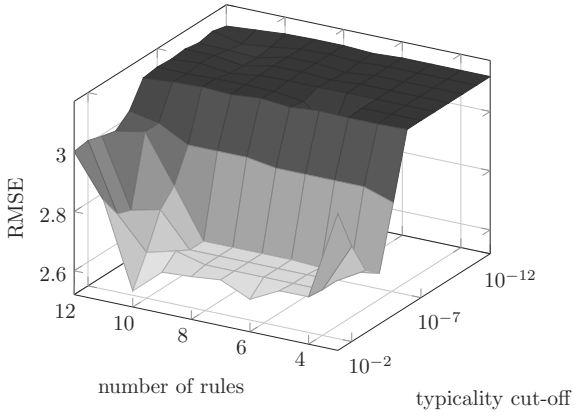

Fig. 16. Root mean square errors for the 'synthetic' data set with the FCOM-ANNBFIS neuro-fuzzy system.

Geng, Y., Li, Q., Zheng, R., Zhuang, F. and He, R. (2018). RECOME: A new density-based clustering algorithm using relative KNN kernel density, *Information Sciences* **436–437**: 13–30.

Grzegorzewski, P., Hryniewicz, O. and Romaniuk, M. (2020). Flexible resampling for fuzzy data, *International Journal of Applied Mathematics and Computer Science* **30**(2): 281–297, DOI: 10.34768/amcs-2020-0022.

Haberman, S.J. (1976). Generalized residuals for log-linear models, *Proceedings of the 9th International Biometrics Conference, Boston, USA*, pp. 104–122.

Harifi, S., Khalilian, M., Mohammadzadeh, J. and Ebrahimnejad, S. (2020). Optimizing a neuro-fuzzy system based on nature-inspired emperor penguins colony optimization algorithm, *IEEE Transactions on Fuzzy Systems* **28**(6): 1110–1124.

Hathaway, R.J., Bezdek, J.C. and Hu, Y. (2000). Generalized fuzzy c-means clustering strategies using l_p norm distances, *IEEE Transactions on Fuzzy Systems* **8**(5): 576–582.

Hekimoglu, S., Erdogan, B. and Erenoglu, R. (2015). A new outlier detection method considering outliers as model errors, *Experimental Techniques* **39**(1): 57–68.

Hekimoglu, S. and Koch, K.R. (2000). How can reliability of the test for outliers be measured?, *Allgemeine Vermessungsnachrichten* **7**: 247–253.

Jakubek, S. and Keuth, N. (2006). A local neuro-fuzzy network for high-dimensional models and optimalization, *Engineering Applications of Artificial Intelligence* **19**(6): 705–717.

Jang, J.-S.R. (1993). ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man, and Cybernetics* **23**(3): 665–684.

Jiang, Y. and Yin, S. (2019). Recent advances in key-performance-indicator oriented prognosis and diagnosis with a Matlab toolbox: DB-KIT, *IEEE Transactions on Industrial Informatics* **15**(5): 2849–2858.

Jiang, Y., Yin, S. and Kaynak, O. (2018). Data-driven monitoring and safety control of industrial cyber-physical systems: Basics and beyond, *IEEE Access* **6**: 47374–47384.

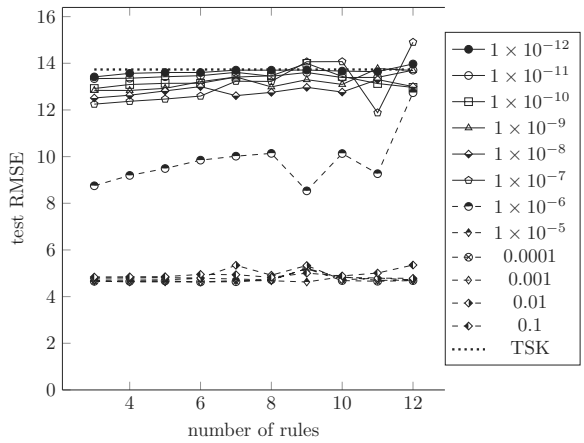

Fig. 17. Root mean square errors for the 'power' data set with the FCOM-TSK neuro-fuzzy system.

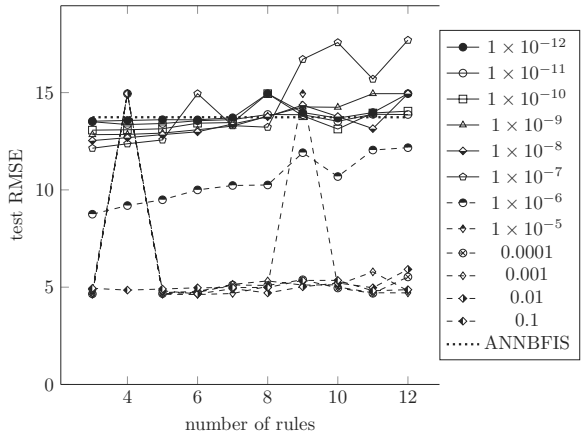

Fig. 19. Root mean square errors for the 'power' data set with the FCOM-ANNBFIS neuro-fuzzy system.

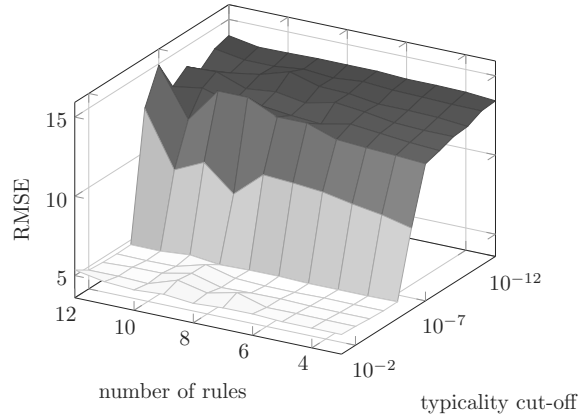

Fig. 18. Root mean square errors for the 'power' data set with the FCOM-TSK neuro-fuzzy system.

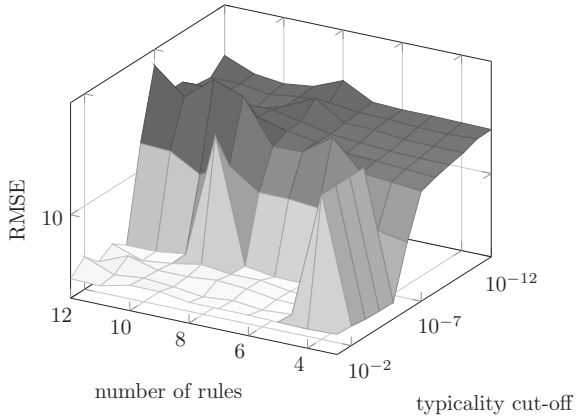

Fig. 20. Root mean square errors for the 'power' data set with the FCOM-ANNBFIS neuro-fuzzy system.

Johnson, B., Tateishi, R. and Hoan, N. (2013). A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees, *International Journal of Remote Sensing* **34**(20): 6969–6982.

Kaya, H., Tüfekci, P. and Gürgen, S.F. (2012). Local and global learning methods for predicting power of a combined gas and steam turbine, *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE 2012), Dubai, UAE*, pp. 13–18.

Keith, M.J., Jameson, A., van Straten, W., Bailes, M., Johnston, S., Kramer, M., Possenti, A., Bates, S.D., Bhat, N.D.R., Burgay, M., Burke-Spolaor, S., D'Amico, N., Levin, L., McMahon, P.L., Milia, S. and Stappers, B.W. (2010). The high time resolution universe pulsar survey. I: System configuration and initial discoveries, *Monthly Notices of the Royal Astronomical Society* **409**(2): 619–627.

Kłopotek, R., Kłopotek, M. and Wierzchoń, S. (2020). A feasible k-means kernel trick under non–Euclidean feature space, *International Journal of Applied Mathematics and Computer Science* **30**(4): 703–715, DOI: 10.34768/amcs-2020-0052.

Krishnapuram, R. and Keller, J. (1993). A possibilistic approach to clustering, *IEEE Transactions on Fuzzy Systems* **1**(2): 98–110.

Latecki, L.J., Lazarevic, A. and Pokrajac, D. (2007). Outlier detection with kernel density functions, *in* P. Perner (Ed.), *Machine Learning and Data Mining in Pattern Recognition*, Springer, Berlin/Heidelberg, pp. 61–75.

Lehmann, R. (2013). 3σ-Rule for outlier detection from the viewpoint of geodetic adjustment, *Journal of Surveying Engineering* **139**(4): 157–165.

Leski, J. and Kotas, M. (2015). On robust fuzzy c-regression models, *Fuzzy Sets and Systems* **279**: 112–129.

Leski, J.M. (2014). Fuzzy c-ordered-means clustering, *Fuzzy Sets and Systems* **286**: 114–133.

Leski, J.M. (2015). Fuzzy $(c + p)$-means clustering and its application to a fuzzy rule-based classifier: Towards good generalization and good interpretability, *IEEE Transactions on Fuzzy Systems* **23**(4): 802–812.

Leski, J.M. and Kotas, M.P. (2018). Linguistically defined clustering of data, *International Journal of Applied Math-*

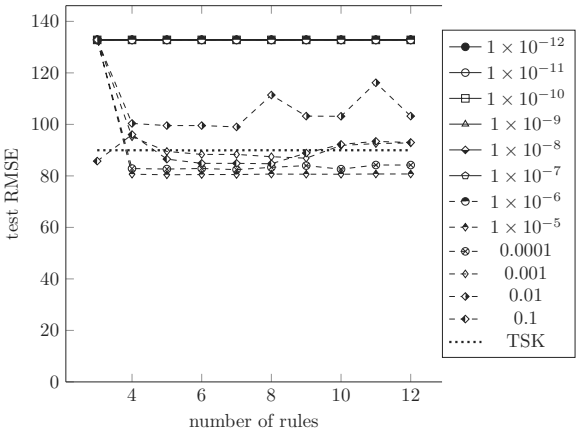

Fig. 21. Root mean square errors for the 'beijing' data set with the FCOM-TSK neuro-fuzzy system.

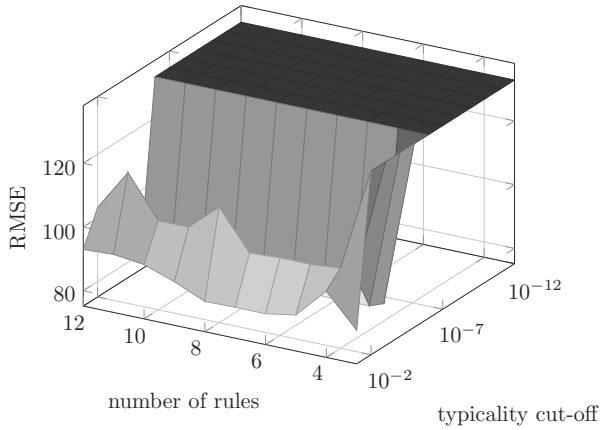

Fig. 22. Root mean square errors for the 'beijing' data set with the FCOM-TSK neuro-fuzzy system.

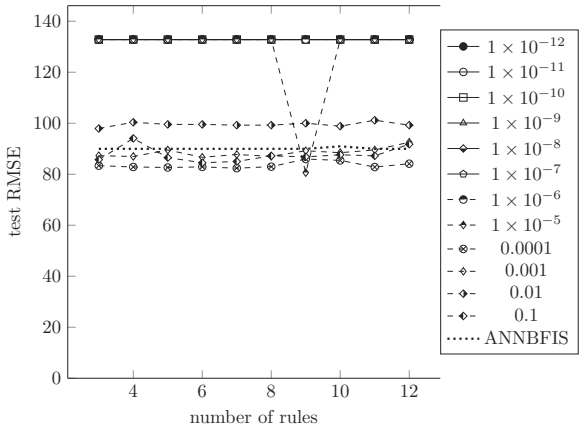

Fig. 23. Root mean square errors for the 'beijing' data set with the FCOM-ANNBFIS neuro-fuzzy system.

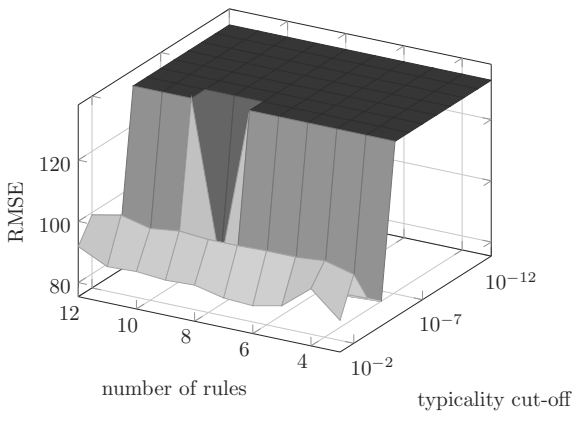

Fig. 24. Root mean square errors for the 'beijing' data set with the FCOM-ANNBFIS neuro-fuzzy system.

ematics and Computer Science **28**(3): 545–557, DOI: 10.2478/amcs-2018-0042.

Leys, C., Ley, C., Klein, O., Bernard, P. and Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median, *Journal of Experimental Social Psychology* **49**(4): 764–766.

Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H. and Chen, S.X. (2015). Assessing Beijing's PM2.5 pollution: Severity, weather impact, apec and winter heating, *Proceedings of the Royal Society A* **471**(2182): 257–276.

Lyon, R.J., Stappers, B.W., Cooper, S., Brooke, J.M. and Knowles, J.D. (2016). Fifty years of pulsar candidate selection: From simple filters to a new principled real-time classification approach, *Monthly Notices of the Royal Astronomical Society* **459**(1): 1104–1123.

Mamdani, E.H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies* **7**(1): 1–13.

Matthews, S.G., Gongora, M.A. and Hopgood, A.A. (2013). Evolutionary algorithms and fuzzy sets for discovering temporal rules, *International Journal of Applied Mathematics and Computer Science* **23**(4): 855–868, DOI: 10.2478/amcs-2013-0064.

Nowicki, R. (2006). Rough-neuro-fuzzy system with MICOG defuzzification, *2006 IEEE International Conference on Fuzzy Systems, Vancouver, Canada*, pp. 1958–1965.

Olson, C.C., Judd, K.P. and Nichols, J.M. (2018). Manifold learning techniques for unsupervised anomaly detection, *Expert Systems with Applications* **91**: 374–385.

Otte, C. (Ed.) (2013). Safe and interpretable machine learning: A methodological review, *in* C. Moewes and A. Nürnberger (Eds), *Computational Intelligence in Intelligent Data Analysis*, Springer, Berlin/Heidelberg, pp. 111–122.

Piegat, A. and Dobryakova, L. (2020). A decomposition approach to type 2 interval arithmetic, *International Journal of Applied Mathematics and Computer Science* **30**(1): 185–201, DOI: 10.34768/amcs-2020-0015.

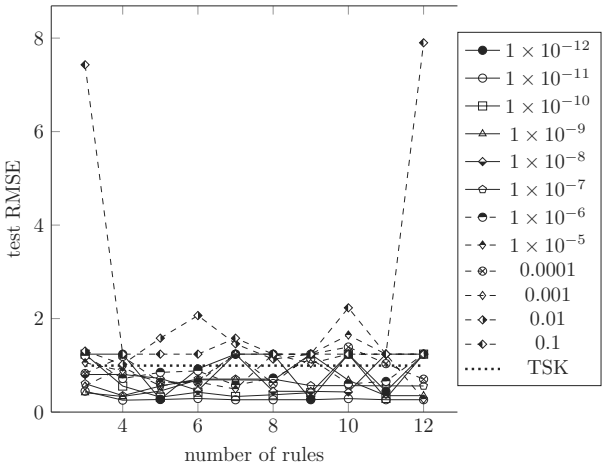

Fig. 25. Root mean square errors for the 'CO_2' data set with the FCOM-TSK neuro-fuzzy system.

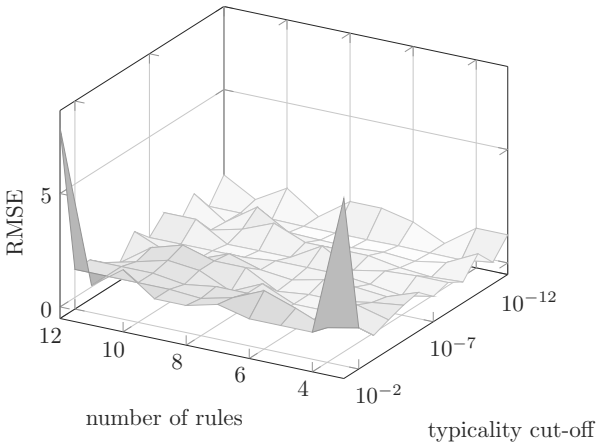

Fig. 26. Root mean square errors for the 'CO_2' data set with the FCOM-TSK neuro-fuzzy system.

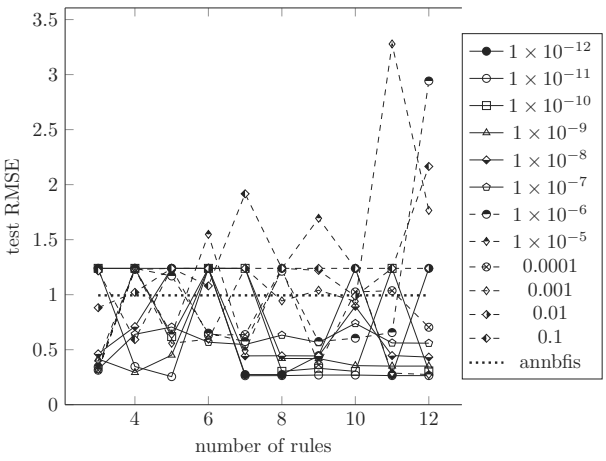

Fig. 27. Root mean square errors for the 'CO_2' data set with the FCOM-ANNBFIS neuro-fuzzy system.

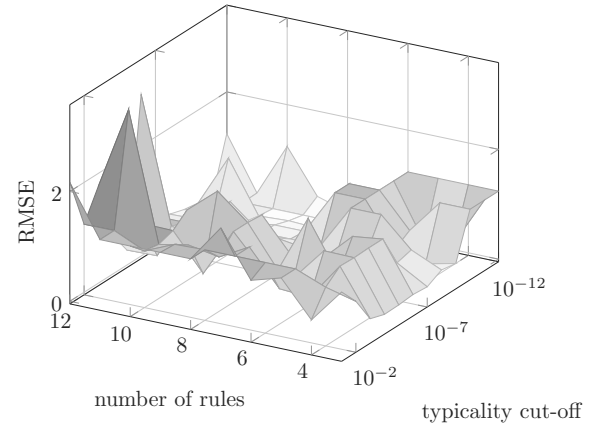

Fig. 28. Root mean square errors for the 'CO_2' data set with the FCOM-ANNBFIS neuro-fuzzy system.

Reichenbach, H. (1935). Wahrscheinlichkeitslogik, *Erkenntnis* **5**: 37–43.

Riid, A. (2002). *Transparent Fuzzy Systems: Modelling and Control*, PhD dissertation, Tallinn Technical University, Tallinn.

Rocha Neto, A.R. and Barreto, G.A. (2009). On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis, *IEEE Latin America Transactions* **7**(4): 487–496.

Seresht, N.G. and Fayek, A.R. (2020). Neuro-fuzzy system dynamics technique for modeling construction systems, *Applied Soft Computing* **93**: 106400.

Sholla, S., Mir, R.N. and Chishti, M.A. (2020). A neuro fuzzy system for incorporating ethics in the Internet of things, *Journal of Ambient Intelligence and Humanized Computing* **12**: 1487–1501.

Sikora, M. and Krzykawski, D. (2005). Application of data exploration methods in analysis of carbon dioxide emission in hard-coal mines, dewater pump stations, *Mechanizacja i Automatyzacja Górnictwa* **413**(6): 57–67.

Siminski, K. (2008). Neuro-fuzzy system with hierarchical domain partition, *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA 2008), Vienna, Austria*, pp. 392–397.

Siminski, K. (2009). Patchwork neuro-fuzzy system with hierarchical domain partition, *in* M. Kurzyński and M. Woźniak (Eds), *Computer Recognition Systems 3*, Advances in Intelligent and Soft Computing, Vol. 57, Springer-Verlag, Berlin/Heidelberg, pp. 11–18.

Simiński, K. (2010). Rule weights in a neuro-fuzzy system with a hierarchical domain partition, *International Journal of Applied Mathematics and Computer Science* **20**(2): 337–347, DOI: 10.2478/v10006-010-0025-3.

Simiński, K. (2012). Neuro-rough-fuzzy approach for regression modelling from missing data, *International Journal of Ap-*

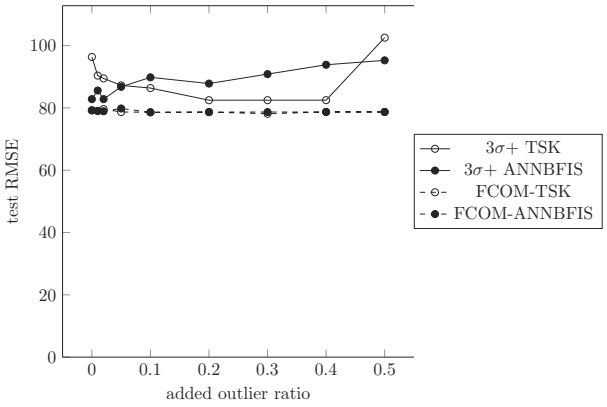

Fig. 29. Comparison of the proposed approach with the 3σ preprocessing technique for various ratios of outliers in the 'beijing' data set.

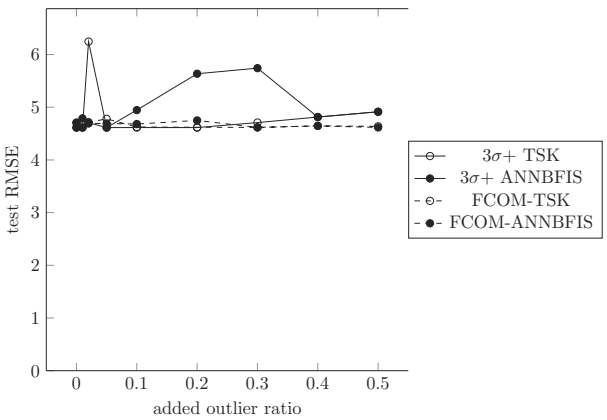

Fig. 30. Comparison of the proposed approach with the 3σ preprocessing technique for various ratios of outliers in the 'power' data set.

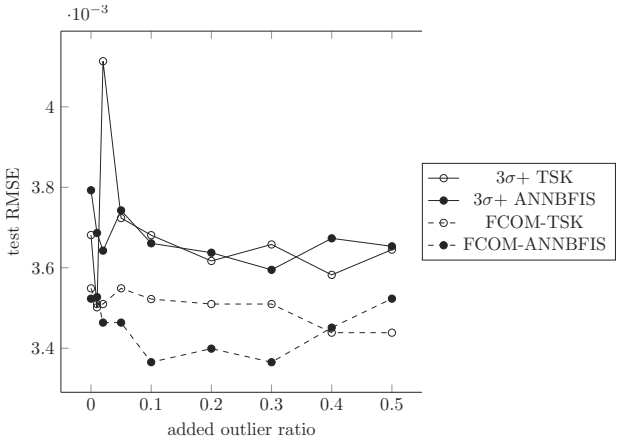

Fig. 31. Comparison of the proposed approach with the 3σ preprocessing technique for various ratios of outliers in the 'carbon' data set.

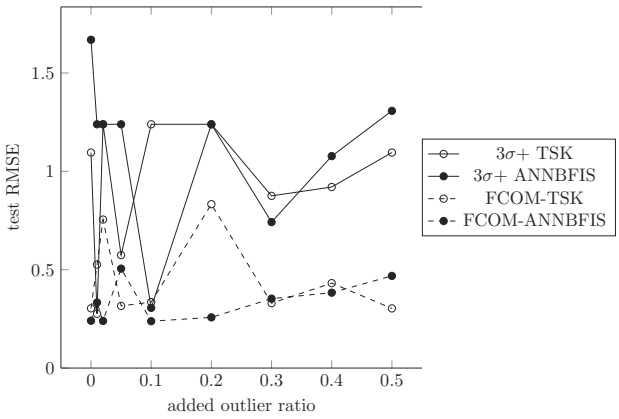

Fig. 32. Comparison of the proposed approach with the 3σ preprocessing technique for various ratios of outliers in the 'CO_2' data set.

plied Mathematics and Computer Science **22**(2): 461–476, DOI: 10.2478/v10006-012-0035-4.

Siminski, K. (2014). Neuro-fuzzy system based kernel for classification with support vector machines, *in* A. Gruca *et al.* (Eds), *Man–Machine Interactions 3*, Springer International Publishing, Cham, pp. 415–422.

Siminski, K. (2015). Rough subspace neuro-fuzzy system, *Fuzzy Sets and Systems* **269**: 30–46.

Siminski, K. (2016). Memetic neuro-fuzzy system with Big-Bang-Big-Crunch optimisation, *in* A. Gruca *et al.* (Eds), *Man–Machine Interactions 4*, Springer International Publishing, Cham, pp. 583–592.

Siminski, K. (2017a). Fuzzy weighted c-ordered means clustering algorithm, *Fuzzy Sets and Systems* **318**: 1–33.

Siminski, K. (2017b). Interval type-2 neuro-fuzzy system with implication-based inference mechanism, *Expert Systems with Applications* **79C**: 140–152.

Siminski, K. (2017c). Robust subspace neuro-fuzzy system with data ordering, *Neurocomputing* **238**: 33–43.

Siminski, K. (2019). NFL—Free library for fuzzy and neuro-fuzzy systems, *in* S. Kozielski *et al.* (Eds), *Beyond Databases, Architectures and Structures: Paving the Road to Smart Data Processing and Analysis*, Springer International Publishing, Cham, pp. 139–150.

Siminski, K. (2020). FIT2COMIn—Robust clustering algorithm for incomplete data, *in* A. Gruca *et al.* (Eds), *Man–Machine Interactions 6*, Springer International Publishing, Cham, pp. 99–110.

Škrjanc, I., Iglesias, J.A., Sanchis, A., Leite, D., Lughofer, E. and Gomide, F. (2019). Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey, *Information Sciences* **490**: 344–368.

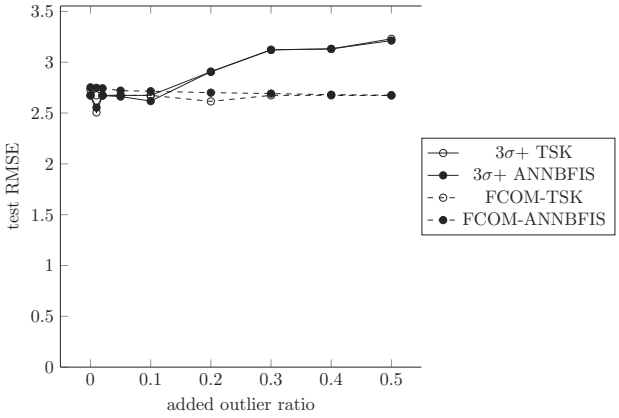

Fig. 33. Comparison of the proposed approach with the 3σ preprocessing technique for various ratios of outliers in the 'synthetic' data set.

Słowik, A., Cpałka, K. and Łapa, K. (2020). Multipopulation nature-inspired algorithm (MNIA) for the designing of interpretable fuzzy systems, *IEEE Transactions on Fuzzy Systems* **28**(6): 1125–1139.

Sugeno, M. and Kang, G.T. (1988). Structure identification of fuzzy model, *Fuzzy Sets and Systems* **28**(1): 15–33.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics* **15**(1): 116–132.

Tang, B. and He, H. (2017). A local density-based approach for outlier detection, *Neurocomputing* **241**: 171–180.

Timm, H., Borgelt, C., Döring, C. and Kruse, R. (2004). An extension to possibilistic fuzzy cluster analysis, *Fuzzy Sets and Systems* **147**: 3–16.

Tüfekci, P. (2014). Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods, *International Journal of Electrical Power & Energy Systems* **60**: 126–140.

Yager, R.R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking, *IEEE Transactions on Systems, Man, and Cybernetics* **18**(1): 183–190.

Yang, P., Zhu, Q. and Zhong, X. (2009). Subtractive clustering based RBF neural network model for outlier detection, *Journal of Computers* **4**(8): 755–762.

Yeh, I.-C., Yang, K.-J. and Ting, T.-M. (2009). Knowledge discovery on RFM model using Bernoulli sequence, *Expert Systems with Applications* **36**(3): 5866–5871.

Youden, W.J. (1950). Index for rating diagnostic tests, *Cancer* **3**(1): 32–35.

Zadeh, L.A. (1973). Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-3**(1): 28–44.

Krzysztof Siminski holds a DSc from the Faculty of Automatic Control, Electronics and Computer Science of the Silesian University of Technology (Gliwice, Poland), where he works at present. His main scientific interests focus on data analysis, machine learning, data mining, granular computing, and formal languages.

Received: 9 November 2020
Revised: 21 January 2021
Re-revised: 5 February 2021
Accepted: 9 February 2021