Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

ABARC: An agent-based rough sets clustering algorithm

Radu D. Găceanu *,1,a, Arnold Szederjesi-Dragomir 2,a, Horia F. Pop 3,a, Costel Sârbu 4,b

[a] Department of Computer Science, Babeş-Bolyai University, str. Mihail Kogalniceanu nr. 1, Cluj-Napoca, 400084, Romania
[b] Department of Chemistry, Babeş-Bolyai University, str. Arany János nr. 11, Cluj-Napoca, 400028, Romania

A R T I C L E I N F O

Keywords:
Clustering
Agent
Rough sets
Overlapping clusters
Outliers

A B S T R A C T

Clustering is an important task in pattern recognition with many applications in natural sciences and healthcare. However, in practical scenarios, it is often the case that the data cannot be easily separated into well distinguished groups for several reasons like: the shape of clusters, the presence of outliers, or the overlapping clusters problem (instances that may belong to more than one cluster). In order to handle such issues, we propose an agglomerative clustering approach which identifies instances that may belong to more than one cluster and clearly separates the outliers form the rest of the instances by integrating concepts from rough sets theory. The whole grouping and regrouping process is driven by software agents executing in parallel. Our approach is computational friendly and experiments on standard data sets indicate its advantages.

1. Introduction

The problem of classification and clustering resides in partitioning a data set into groups, one group for each category of input. Hence, it attempts to assign every instance to one of the possible classes or clusters. In real life, the complete description of the classes is not known although a finite and usually smaller number of instances are available (training set), which often provide partial information for the optimal design of the classification or clustering system. The parameter values of various pattern recognition models are decided based on the information from the training set. Clustering is an unsupervised learning problem and it deals with partitioning an unlabeled data set in such a way that two objects from the same group are as similar as possible, while objects from different groups are as dissimilar as possible. Classification and clustering are nowadays applied in a wide range of activity domains including healthcare in general, image recognition, fake news detection, sentiment analysis just to name a few (Onan, 2018a; 2018b; 2019a; 2019b; 2019c; 2020; 2021a; 2021b; 2022; Onan and Korukoğlu, 2017; Onan et al., 2016; 2017; Onan and Toçoğlu, 2021).

There are a number of characteristics to be taken into account when designing a clustering algorithm. Many algorithms work well on small data sets, but when applied on larger ones (image recognition, web mining, gene expression data) they may be impractical. This is why scalability is an important aspect to be taken into account in clustering and classification. Other issues to be considered are: the ability of discovering clusters of arbitrary shape and density, handling overlapping clusters, dealing with noisy data. An important property of a clustering system, especially from an end-user perspective, is the interpretability of the results (Han et al., 2011).

One of the important problems in real-life data analysis is uncertainty management which, in a clustering scenario, includes the problems of overlapping clusters, clusters of arbitrary shape, and noisy data. In this regard, the concepts of fuzzy sets theory (Zadeh, 1965) and rough sets theory (Pawlak, 1992) have been applied in order to deal with impreciseness and vagueness.

Probably the most widely used clustering algorithm is k-Means (Macqueen, 1967), where each object is assigned to exactly one cluster. But it is often the case that clusters are not well defined and this is why the problem may be reformulated in terms of fuzzy or rough clusters. In the Fuzzy c-Means algorithm (FCM) (Bezdek, 1981), instances may belong to more than one cluster according to a membership degree. The FCM algorithm can deal with overlapping clusters, but it may be

* Corresponding author.
E-mail addresses: radu.gaceanu@ubbcluj.ro (R.D. Găceanu), arnold.szederjesi@ubbcluj.ro (A. Szederjesi-Dragomir), horia.pop@ubbcluj.ro (H.F. Pop), costel.sarbu@ubbcluj.ro (C. Sârbu).
1 [orcid=0000-0002-0977-4104]
2 [orcid=0000-0002-1106-526X]
3 [orcid=0000-0003-2777-7541]
4 [orcid=0000-0001-9374-2078]

https://doi.org/10.1016/j.iswa.2022.200117
Received 25 May 2022; Received in revised form 17 July 2022; Accepted 23 August 2022
Available online 27 August 2022

inaccurate in a noisy environment (Krishnapuram and Keller, 1993). The rough sets theory takes a different approach to dealing with uncertainty by exploiting the concept of indiscernibility with respect to some similarity measure. Lately, rough sets have been applied in many scenarios including: addressing the four dimensional transportation problem (Bera et al., 2018) (using rough intervals), if-then rules induction from decision tables (Kato et al., 2018), decision making (Liu et al., 2018; Pamucar et al., 2018), addressing the minimum weight vertex cover problem (Xie et al., 2018), stock price prediction (Lei, 2018), healthcare (Wang et al., 2011; Yang and Wu, 2009) (using feature selection), relational facts categorization (Bharadwaj and Ramanna, 2019). Apparently, rough sets are mostly used for feature selection, while their application in clustering seems to be less common even though there are some interesting rough hybridisations of partitioning clustering algorithms as well. The Rough c-Means algorithm (RCM) (Lingras and West, 2004) is an example in this sense, where clusters are described in terms of a prototype together with a lower and upper approximation. The Rough Possibilistic c-Means algorithm and the generalized form of the c-Means algorithm (Maji and Pal, 2007) are proposed in order to address noisy data. In Chen et al. (2006), a hierarchical clustering algorithm based on rough sets is presented. The authors model the clustering problem using decision tables $(U, A \cup \{d\})$, where U is the universe of discourse (all instances $x^i \in U$), A is the attribute set, and d is an introduced decision attribute. In rough sets theory terms, the attribute d determines an indiscernibility relation which partitions U in equivalence classes. The algorithm is applied for clustering categorical data.

This paper presents a *rough sets* approach to clustering in order to deal with uncertainty in data like *overlapping clusters* or *outliers*. As far as we know, our approach is different from other rough clustering methods in several ways. The whole clustering process is performed by *software agents* (Wooldridge, 2009) that cooperate and self-organize themselves into groups containing similar individuals. Agents are implemented as lightweight processes in the Elixir programming language, yielding a highly scalable solution. Evaluating the quality of overlapping data and outlier discovery abilities of the algorithm proved to be a challenging tasks because of the lack of benchmark datasets for this specific contexts. This is why we propose what we believe to be an *an objective methodology for assessing the quality of a clustering solution in case of overlapping clusters and outliers*. Experiments on standard and synthetic data sets outline the advantages of the proposed approach.

The main contributions of this paper are:

- An agglomerative clustering algorithm using concepts from the rough sets theory and software agents
- The ability to handle overlapping clusters and noisy data
- An objective methodology for assessing the quality of a clustering solution in case of overlapping clusters and outliers
- A scalable clustering approach

The rest of the paper is structured as follows: Section 2 presents the motivation of our work, emphasizing the importance of the proposed approach. Section 3 presents a theoretical background regarding rough sets and formulates the rough sets clustering problem. The main algorithms proposed by us are described in Section 4. The same section presents an objective methodology that we propose for evaluating the clustering results in case of overlapping clusters and outliers, arguing that some of the widely used cluster evaluation measures are biased. Experiments on standard data sets are detailed in Section 5 and a comparison to related approaches is presented in Section 6. Finally, Section 7 draws the conclusions of this paper and presents ideas for future work.

2. Motivation

Clustering is the problem of organizing a collection of elements into coherent groups in such a way that similar elements are in the same cluster and different elements are in different clusters. One of the most popular clustering algorithms is k-Means (Macqueen, 1967), a partitioning clustering method which has many extensions including some recent ones (Bagirov et al., 2016; 2011; Gribel and Vidal, 2019; Ismkhan, 2018). The main drawback of partitioning methods is the fact that the number of clusters has to be known in advance, which is not the case for the hierarchical (agglomerative or divisive) counterparts.

Many algorithms perform well when the clusters are clearly separated, but having non-overlapping clusters in practical scenarios is rather the exception and not the rule. There are several reasons that might cause overlapping: there might be noise in the data, the features may not capture all the necessary information to clearly separate clusters, or the overlap may be inherent to the processes that produced the data (Adam and Blockeel, 2015). The fuzzy sets theory (Zadeh, 1965) has been applied extensively in order to deal with uncertainty and vagueness (Luo et al., 2022; Maneckshaw and Mahapatra, 2022; Sakhardande and Gaonkar, 2022; Shang et al., 2022; Yang et al., 2022), as well as for providing a more natural and detailed data interpretation. For example, in Pop et al. (1996); Sârbu et al. (1996) the chemical elements of the periodic system are classified based on some of their physical, chemical and structural features suggesting in some cases slightly different partitions from the official ones. The fuzziness of the approach allows a much better analysis of the similarities between elements as well as of the properties that are involved in producing these similarities.

A major issue in pattern recognition is outlier management which, if improperly handled, could have a major impact on the result. For example, in a regression analysis context, even a single outlier could greatly influence the outcome as shown in Pop and Sârbu (1996) and in order to tackle this issue a fuzzy regression algorithm is proposed that is able to appreciate the presence of outliers and the linearity of the regression line.

In pattern recognition, the rough sets theory (Pawlak, 1992) addresses the same issues as the fuzzy sets theory does but in a different fashion, by approximating an indistinguishable set in terms of a lower approximation set and an upper approximation set. The lower approximations set contains elements which surely belong to the indistinguishable set, while the upper approximation contains elements that may belong to the indistinguishable set. From a data analyst stand point, this approach may constitute an advantage over a fuzzy sets one since the data that surely belongs to the indistinguishable set is clearly separated, allowing the analyst to focus only on the remaining elements, the ones for which the membership is uncertain. Rough sets were primarily used for feature extraction as shown in Section 1 and in a much lesser extent to directly model clusters. In fact, except our previous work on this matter (Szederjesi-Dragomir et al., 2020; 2019), the only algorithms we know of that use rough sets in order to represent clusters are the ones from Li et al. (2019); Lingras and West (2004); Maji and Pal (2007, 2012a) which are all partitioning based methods. On the other hand, *our algorithm is a hierarchical one, it is able to detect outliers and overlapping clusters by using rough sets*. Moreover, by employing software agents (Wooldridge, 2009) which are able to execute in parallel, it is also *scalable*.

3. Problem statement

Pattern recognition is an activity that normally human beings excel in, but it is not a trivial task for a computer program due to the uncertainty involved in the entire process. Sources of uncertainty include incomplete data, vagueness in class definitions and the presence of outliers. The fuzzy sets (Zadeh, 1965) and rough sets (Pawlak, 1992) theories have been applied in order to handle uncertainty at different levels of a pattern recognition process. Even though both theories are developed in order to address vaguness, they are in general complementary because while fuzzy sets address the gradualness of knowledge (expressed by the membership degree), rough sets address the granularity of knowledge (expressed by the indiscernibility relation) (Maji and Pal, 2012b).

3.1. Rough sets

The rough sets theory (Pawlak, 1992) is a major mathematical instrument for managing uncertainty and it consists in offering a formal approximation of a crisp set in terms of a pair of sets representing the lower and upper bound of the original set.

Given a set of objects U called the universe of discourse, an equivalence relation $R \subseteq U \times U$, and a subset of U denoted by X, in order to approximate X with respect to R we consider the following definitions:

Definition 1. The **lower approximation** of a set X with respect to R is the set of all objects which **certainly** belong to X: $\underline{R}X = \bigcup_{x\in U}\{R(x) : R(x) \subseteq X\}$.

Definition 2. The **upper approximation** of a set X with respect to R is the set of all objects which **possibly** belong to X: $\overline{R}X = \bigcup_{x\in U}\{R(x) : R(x) \cap X \neq \varnothing\}$.

Remark 1. The lower and upper approximation sets are crisp.

Definition 3. The **boundary region** of a set X with respect to R is the set of all objects which can not certainly be classified as either belonging to X or not belonging to X: $RB = \overline{R}X - \underline{R}X$.

Remark 2. If the **boundary region** is empty then the set X is crisp, i.e., precise with respect to R.

Definition 4. A **rough set** is a tuple $\langle \underline{R}X, \overline{R}X\rangle$, where $\underline{R}X$ is the **lower approximation** (Definition 1) of the target set X and $\overline{R}X$ is the **upper approximation** (Definition 2) of the target set X.

3.2. Rough sets clustering

Cluster analysis is a technique of finding natural groups in a data set. The idea may be formalized as in Definition 5.

Definition 5. **Clustering** is the process of finding a set $C = \{C_k | k = \overline{1,p}\}$ of subsets of a given set of objects $X = \{x^i | i = \overline{1,n}, 1 \leq p \leq n\}$ such that:

1 $\forall k,l = \overline{1,p}, k \neq l : C_k \cap C_l = \varnothing$
2 $\bigcup_{k=1}^{n} C_k = X$
3 $\forall k = \overline{1,p} : C_k \neq \varnothing$
4 $\forall k = \overline{1,p}, \forall i,j = \overline{1,|C_k|} : sim(x^i, x^j)$ − holds,

where: $sim(x^i, x^j)$ denotes the similarity between two items x^i and x^j.

Remark 3. The set C is a partition of the given set of objects X (see Properties 1, 2, and 3 from Definition 5) such that the objects from each subset C_k are similar with each other (see Property 4).

Let $\underline{R}C_K$ and $\overline{R}C_k$ denote the lower and upper approximations of a cluster C_k, and let $RB = \overline{R}C_k - \underline{R}C_k$ be the boundary region of C_k. The tuple $\langle \underline{R}C_k, \overline{R}C_k\rangle$ is called the rough set associated to C_k with respect to some equivalence relation R (see Definition 4). Then the rough sets clustering problem may be defined as in Definition 6.

Definition 6. **Rough sets clustering** is the process of finding a set $RC = \{\langle \underline{R}C_k, \overline{R}C_k\rangle | k = \overline{1,p}\}$ of subsets of a given set of objects $X = \{x^i | i = \overline{1,n}, 1 \leq p \leq n\}$ such that:

- $\forall k,l = \overline{1,p}, k \neq l : \underline{R}C_k \cap \underline{R}C_l = \varnothing$
- $\forall k,l = \overline{1,p}, k \neq l : |\overline{R}C_k \cap \overline{R}C_l| \geq 0$
- $\bigcup_{k=1}^{n} \overline{R}C_k = X$
- $\forall k = \overline{1,p} : \underline{R}C_k \neq \varnothing$
- $\forall k = \overline{1,p} : \underline{R}C_k \subseteq \overline{R}C_k$
- $\forall k = \overline{1,p}, \forall i,j = \overline{1,|\underline{R}C_k|} : x^i R x^j \wedge x^i, x^j \in C_k$
- $\forall k = \overline{1,p}, \forall i,j = \overline{1,|\overline{R}C_k|} : x^i R x^j \wedge |\{x^i, x^j\} \cap C_k| \geq 0.$

Remark 4. Based on Definition 6, the following properties hold:

- $x^i \in \underline{R}C_k, i = \overline{1,n}, k = \overline{1,p} \Longrightarrow x^i \notin \underline{R}C_l, \forall l = \overline{1,p}, k \neq l$
- $x^i \in \underline{R}C_k \Longrightarrow x^i \in \overline{R}C_k, i = \overline{1,n}, k = \overline{1,p}$
- $x^i \notin \underline{R}C_k, i = \overline{1,n}, \forall k = \overline{1,p} \Longrightarrow \exists l, l = \overline{1,p} \ni x^i \in \overline{R}C_l.$

So in a rough sets clustering approach, every cluster C_k is defined in terms of a lower approximation and an upper approximation. The instances from the lower approximation certainly belong to the cluster C_k with respect to some similarity measure. The instances from the upper approximation possibly belong to C_k, but we cannot be certain about this, according to the considered similarity measure. If an instance belongs to a lower approximation then it does not belong to any other rough set. On the other hand, if an instance is in the boundary region of a cluster C_k then it may belong to several other boundary regions. Also, if an instance belongs to the lower approximation of a cluster then it also belongs to the upper approximation of that cluster.

The whole idea may be visualized in Fig. 1, where two not clearly separable clusters are shown. Even though they can not be precisely identified, the two clusters may still be described by their lower approximations (the ellipses) and by their upper approximations (the rectangles). In this example the boundary regions are overlapping and the red instances from the image belong to both upper approximations and hence, to some extent, to both clusters.

4. Proposed approach

This section presents our approach to clustering. We introduce **ABARC** (**A**gent **BA**sed **R**ough **C**lustering), an algorithm which addresses the overlapping clusters problem by modeling clusters using notions from the rough sets theory. The algorithm successfully identifies outliers and, by using software agents, it is also scalable. Section 4.1 provides a detailed description of the algorithm together with an example that better motivates our approach. In Section 4.2 we introduce what we believe to be an objective methodology for evaluating the quality of a clustering result in case of overlapping clusters and outliers. We also discuss about some widely used cluster evaluation measures (Section 4.2.2) and we argue that they may be biased in some cases.

4.1. Agent-based clustering algorithm using rough sets

This section presents the ABARC algorithm, our approach to rough

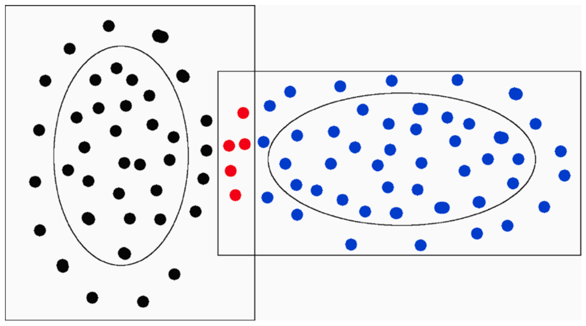
Fig. 1. Rough clusters with overlapping upper approximations.

sets clustering using software agents. The main clustering procedure is described in Algorithm 1, where X is the data set containing the instances to be clustered, $i_{\max}$ and λ are integers denoting the number of trials for specific tasks (details bellow), σ_1 is the similarity limit and δ is a distance metric (for example, the Euclidean distance). The similarity limit, σ_1, is specific to every data set and it is a real number denoting the maximum value up to which two instances are considered similar (they surely belong to the same group). The first step is to initialize the agents and associate one agent to each instance. Also, each agent is assigned to a different cluster, so, at the beginning, the number of clusters is equal to the number of agents which is equal to the number of instances. Agents are executed in parallel, in separate processes, and this behavior is indicated by the $\|$ operator from line 4.

Algorithm 2 shows the asynchronous behavior of each agent: given an *agent*$_k$, it tries to find similar fellows with respect to σ_1 and δ by direct message exchange. One it finds a similar agent, it moves to its cluster (line 3).

Algorithm 3 describes the procedure of finding a similar agent. The argument λ denotes the maximum number of attempts the agent should perform in order to find a similar one. On line 4, an agent is selected in a non-deterministic way as indicated by the $\prod$ operator and, if the two agents are not in the same cluster, their *similarity* is computed. If this value is bellow the similarity limit, σ_1, then a similar agent is found and the function terminates by returning the *selectedAgent*. Otherwise the search continues and after λ failed attempts of finding similar agents (which are not located in the current cluster) the function returns *null* meaning that either there are no agents similar to the given one (*agent*$_k$) or the process simply took too much time in which case the task is left to other agents or to another iteration (line 2 from Algorithm 1) when the search process is probably faster since there are less clusters. The *computeSimilarity* function from line 6 computes the similarity between two agents given a distance metric δ. If this value is bellow the similarity limit, σ_1, it means that the two agents *certainly* belong to the same cluster so they are in the *lower approximation* of the current cluster.

Algorithm 4 represents the second phase of our approach. Since agents are grouped together only if they should certainly belong to the same cluster (based on the similarity limit, σ_1), the first phase of our approach (Algorithms 1,2 and 3) will probably produce a large number of clusters. The second phase (Algorithm 4) unifies similar clusters producing rough clusters. The algorithm receives as a first argument a set of cluster representatives. A *cluster representative*, $\mathscr{R}_k = \langle C_k, \mathscr{A}(C_k)\rangle$, is a tuple where $\mathscr{A}(C_k)$ is the centroid of the cluster C_k and it is computed as follows:

$$\mathscr{A}(C_k) = \frac{1}{|C_k|}\sum_{x^i \in C_k} x^i \tag{1}$$

The second parameter, σ_2, is a *rough* similarity limit denoting up to what point two agents are *possibly* similar. This parameter is different from the σ_1 value (used in Algorithms 1,2 and 3) which denotes the point up to which two agents are *surely* similar. The last argument, *unified*, denotes the set of unified clusters and it is initially equal to the empty set. Cluster similarity is computed based on the centroid values in the same fashion agent similarity is performed. If a representative is similar to several ones then the corresponding data will belong to several clusters in the upper approximation. The result is a set of representatives of the unified clusters. The *updateRepresentatives* function from line 8 recalculates the representatives using Eq. 1.

Even after executing Algorithm 4 there might be a significant number of clusters remaining, but most of them are normally composed of a very small number of entities which are not similar to either of the 'normal' clusters. The instances from these small clusters will be marked as possible *outliers*. Nevertheless, in Algorithm 5, the third phase of our approach, we will assign them to the closest cluster and we get the final clustering structure.

Algorithm 5 receives as input data the *clusters*, as resulted after

Algorithm 1. Agent Clustering.

```
Data: $X, i_{\max}, \lambda, \sigma_1, \delta$
Result: $RC$ //the set of rough clusters
1 @ Let $\mathcal{AG}$ be the set of agents
2 for $i = \overline{1, i_{\max}}$ do
3     for $k = \overline{1, |\mathcal{AG}|}$ do
4         $\| doCluster(agent_k, \lambda, \sigma_1, \delta, \mathcal{AG})$
5     end
6 end
```

```
Data: $agent_k, \lambda, \sigma_1, \delta, \mathcal{AG}$
$sa_k = searchForSimilar(agent_k, \lambda, \sigma_1, \delta, \mathcal{AG})$
if $sa_k \neq null$ then
    $changeCluster(agent_k, sa_k)$
end
```

Algorithm 2. Do Cluster.

```
Data: $agent_k, \lambda, \sigma_1, \delta, \mathcal{AG}$
Result: $sa$ //similar agent
if $\lambda = 0$ then
    return $null$
end
$\prod\{selectedAgent = a_j, \forall j = \overline{1, |\mathcal{A}|}, a_j \in \mathcal{AG}\}$
if $getCluster(agent) \neq getCluster(selectedAgent)$ then
    $similarity = computeSimilarity(agent, selectedAgent, \delta)$
    if $similarity \leq \sigma_1$ then
        return $selectedAgent$
    else
        return $searchForSimilar(agent, \lambda - 1, \sigma_1, \delta, \mathcal{AG})$
    end
else
    return $searchForSimilar(agent, \lambda - 1, \sigma_1, \delta, \mathcal{AG})$
end
```

Algorithm 3. Search for Similar.

```
   **Data**: $representatives, \sigma_2, unified$
   **Result**: $unified$
1  **if** $representatives = \emptyset$ **then**
2      **return** $unified$
3  **end**
4  $\mathcal{R}_k = first(representatives)$
5  $\mathcal{S} = getSimilar(\mathcal{R}_k, representatives \setminus \{\mathcal{R}_k\}, \sigma_2)$
6  **if** $\mathcal{S} \neq \emptyset$ **then**
7      $updateCluster(\mathcal{R}_k\langle C_k\rangle, \mathcal{S})$
8      $newRepresentatives = updateRepresentatives(representatives)$
9      **return** $SimilarClusterUnification(newRepresentatives, \sigma_2, \emptyset)$
10 **else**
11     **return** $SimilarClusterUnification(representatives \setminus \{\mathcal{R}_k\}, \sigma_2, unified \cup \{R_k\})$
12 **end**
```

Algorithm 4. SimilarClusterUnification.

```
   **Data**: $clusters, \varepsilon$
   **Result**: $finalClusters$
1  $\{outliers, clusters\} = detectOutliers(clusters, \varepsilon)$
2  $finalClusters = joinOutliers(outliers, clusters)$
3  **return** $finalClusters$
```

Algorithm 5. Outlier Elimination.

applying Algorithm 4. In line 1, the outliers (which are themselves clusters) are separated from the 'normal' clusters. This decision is based on the value of ε and the cluster size: if the number of instances from a cluster is less then ε then the cluster is marked as an outlier. The value of ε is set to 5% from the total number of instances in the data set. In line 2, each outlier is unified with the closest 'normal' cluster, based on the cluster representatives.

4.1.1. *Example*

In order to better explain and motivate our approach, we consider a synthetic data set containing ten instances with two attributes each, as shown in Table 1. This example does not illustrate all the steps of the algorithm, it is rather meant to show one possible scenario where it could be more useful compared to other approaches.

Instances from 1 to 4 with attribute values between 0 and 0.25 are very similar with each other and should clearly belong to one cluster. The same observation is valid for instances from 5 to 8 with attribute values between 0.75 and 1 which should clearly belong to a different cluster. These two clusters should be well separated from each other since instances from the first group $(1-4)$ are very different with respect to instances from the second group $(5-8)$. The problem lies with instances 9 and 10 whose attribute values (between 0.4 and 0.6) are close enough to some instances from both groups. The whole idea may be visualized in Fig. 2, where instances 9 and 10 are colored in green.

The first phase of our clustering approach (Algorithms 1 and 3) will actually produce three clusters: $C_1=\{1,2,3,4\}$, $C_2=\{5,6,7,8\}$ and $C_3=\{9,10\}$. The representatives (see Algorithm 4) of these clusters are: $RC_1=\{0.1375,0.1375\}$, $RC_2=\{0.8625,0.8625\}$ and $RC_3=\{0.45,0.55\}$. After the second phase of our clustering approach the data from cluster C_3 will be unified with the data from clusters C_1 and C_2: instance 9 will be added to both C_1 and C_2 because the similarity between instance 9 and cluster representatives RC_1 and RC_2 is 0.26 in both cases which is bellow the rough similarity limit, 0.3; instance 10 will be added to both C_1 and C_2 because the similarity between instance 10 and clusters representatives RC_1 and RC_2 is 0.28 in both cases which is also bellow the rough similarity limit. The squared Euclidean distance was used for computing the similarity between two instances. Thus the final result of our approach is two clusters: $C_1=\{1,2,3,4,\mathit{9},\mathit{10}\}$, $C_2=\{5,6,7,8,\mathit{9},\mathit{10}\}$, where instances 9 and 10 belong to the upper approximation of both clusters and are clearly marked as such.

4.2. *Proposed methodology*

We consider several data sets for our experiments and this section outlines the steps that we perform for each case study. Since our algorithm outputs a set of rough clusters, i.e., some instances are assigned to upper approximations, we need a mean to asses the reliability of this result. We perform a validation step for each case study in which we compare what we report as hybrid data with the actual (as per the data set documentation) clusters. The obtained results are also evaluated in terms of classifications errors and other widely used metrics in assessing the quality of a clustering process. The validation part is implemented in the Ruby programming language.

Table 1
Synthetic data set .

Id	a1	a2
1	0	0
2	0.1	0.1
3	0.2	0.2
4	0.25	0.25
5	0.75	0.75
6	0.8	0.8
7	0.9	0.9
8	1	1
9	0.5	0.5
10	0.4	0.6

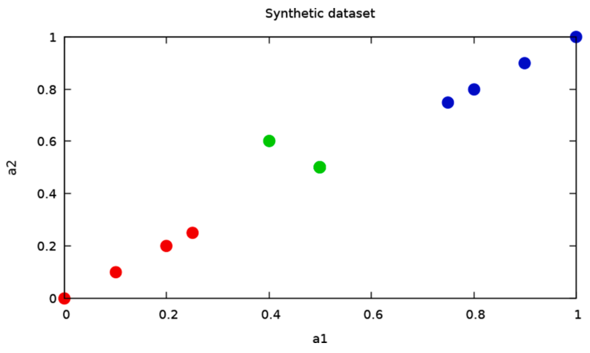

Fig. 2. Synthetic data set with hybrid data.

4.2.1. *Validation*

Standard classification or clustering measures do not take into account the hybrid nature of data: the fact that some instances are marked as being rough, others possible outliers. Even if some metrics have been proposed in this sense, we still would like to be able to claim that some instances identified by us as being rough or outliers are indeed so. This is why we propose an additional analysis for validating the results. We consider several case studies in our experiments, one case study for each data set, and in this section we describe the steps performed within each case study, along with the proposed hybrid data validation methodology.

The steps that we perform in conducting the experiments for each case study are:

1. Execute the algorithm from Section 4.1 for a given data set. The result will be a set of rough clusters, the instances from the upper approximations being clearly outlined.
2. Validate the results
   a Compute cluster representatives (for the clusters reported by the data set documentation). Unless otherwise specified in the experiments, a cluster representative, $\mathscr{R}_k=\langle C_k,\mathscr{A}(C_k)\rangle$, is a tuple where $\mathscr{A}(C_k)$ is the centroid of the cluster C_k and it is computed as follows: $\mathscr{A}(C_k)=\frac{1}{|C_k|}\sum_{x^i\in C_k}x^i$.
   b Compute the standard deviations for the data used for centroid calculation.
   c Identify the most dissimilar instances for each cluster, based on the similarity between the cluster representative and the instances. Unless otherwise specified in the experiments, the Squared Euclidean distance is used in computing the similarity level.
   d Based on the standard deviations computed at Step 2b, we may decide to remove some instances considered to be hybrid from the data set and repeat Steps 2b, 2b and 2b.
   e Compare the hybrid data reported by our algorithm with the instances from Step 2c.
3. Evaluate cluster quality using standard measures, possibly taking into account the analysis from Step 2.

In order to better analyze the hybrid data obtained by our algorithm as well as the classification errors (within Step 2e), we apply an additional procedure and and we introduce Definitions 7, 8, 9 and 10 in this regard.

Definition 7. The **rough threshold**, \boldmath $\mathscr{T}_R^k$, of a cluster C_k is the distance between the most dissimilar instance in cluster C_k and its representative.

Remark 5. The rough threshold is computed after the hybrid data is eliminated, so after performing Step 2d.

Definition 8. An instance x^i from cluster C_j **should be rough** if the following condition is fulfilled:

$$\exists k \ni |d_k^i - \min_k d_k^i| < \mathscr{T}_R^k + \mathscr{T}_R^j, \forall k = \overline{1,p}, k \neq \operatorname{argmin}_k d_k^i \quad (2)$$

where p denotes the number of clusters, and d_k^i is the distance from instance x^i to cluster C_k.

Definition 9. For a fixed ϵ, the **outlier threshold**, \boldmath $\mathscr{T}_O$, of a data set with n instances x^i $(i = \overline{1,n})$ and p clusters C_k $(k = \overline{1,p})$, is given by

$$\underset{\mathscr{T}_O}{\operatorname{argmax}} f(\mathscr{T}_O)$$

subject to

$$\frac{f(\mathscr{T}_O)}{n} < \epsilon$$

where:

$$f(\mathscr{T}_O) = \sum_{k=1}^{p} \left| \left\{ x^i | d_k^i > \mathscr{T}_O, x^i \in C_k, i = \overline{1,n} \right\} \right|.$$

Definition 10. An instance x^i **should be an outlier** if the following condition is met: $d_k^i > \mathscr{T}_O, \forall k = \overline{1,p}$.

We apply the following steps in order to analyze the reported rough instances, outliers and classification errors:

1 Identify the most dissimilar instance from each cluster (see Step 2c). Compute the rough thresholds $\mathscr{T}_R^k$ for all clusters after the hybrid data is eliminated according to Definition 7 and Remark 5.
2 Check all the reported rough instances against Definition 8.
3 Check all the reported outliers against Definition 10. Unless otherwise specified in the experiments, the value of ϵ is set to 0.15. So the value of $\mathscr{T}_O$ is computed such that at most 15% of the instances from the original data set may be considered outliers.
4 Any classification errors will also be analyzed using the information presented in this section.

4.2.2. *Cluster evaluation measures*

In order to evaluate the quality of the obtained clusters we employ the following metrics: *DB* (Davis-Bouldin Index) (Davies and Bouldin, 1979), *DU* (Dunn Index) (Bezdek and Pal, 1998) and *SI* (Silhouette) (Rousseeuw, 1987).

The Davis-Bouldin index is computed based on the ratio between within-cluster distances and inter-cluster distances as follows:

$$DB = \frac{1}{c} \sum_{i=1}^{c} \max_{\substack{1 \leq k \leq c \\ i \neq k}} \left\{ \frac{\Delta_i + \Delta_k}{\delta(C_i, C_k)} \right\},$$

where c represents the number of clusters, Δ_i is the within-cluster distance for the cluster C_i, and $\delta(C_i, C_k)$ represents the inter-cluster distance. A good clustering procedure should make this index as low as possible.

As opposed to the *DB* index which considers the distances between all clusters, the Dunn index only takes into account the minimum inter-cluster distance and the maximum within-cluster distance. The higher the Dunn index, the better the clustering solution is. It is given by the following formula:

$$DU = \frac{\underset{1 \leq i < j \leq c}{min} \delta(C_i, C_k)}{\underset{1 \leq k \leq c}{max} \Delta_k}$$

The Silhouette index measures how similar an instance is to its cluster in comparison to the other clusters and is given by the following formula:

$$SI = \frac{1}{n} \sum_{i=1}^{n} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where n is the number of instances, $a(i)$ is the average distance between instance i and all other instances from the same cluster, and $b(i)$ is the smallest average distance between instance i and all instances from the other clusters. The value of this index lies in the $[-1, 1]$ interval and a good clustering algorithm should make this index as close to 1 as possible.

However, as pointed out in Lamirel et al. (2016), many indexes (including the aforementioned ones) have the drawback to be sensitive to noisy data and outliers. The authors also observed that these indexes are not suitable to analyze clustering results for highly dimensional data as well as for detecting degenerated clustering results. Another issue pointed out in Lamirel et al. (2016) is that these indexes are not independent of the clustering method with which they are used. As an example, a clustering method which tends to optimize the within group sum of squared errors, like k-Means, will also tend to naturally produce good values for that criterion in index computation, but does not necessarily guarantee coherent results (Lamirel et al., 2011). Moreover, when comparing two algorithms based on such indexes, the similarity measures should be the same, otherwise the results may be biased. *This is why we prefer performing this comparison based on the analysis procedure described in Section 4.2.1*.

On the other hand, external evaluation measures like entropy (Shannon, 2001) could be considered:

$$E = \sum_{i=1}^{c} \frac{|C_i|}{n} E(C_i),$$

where c represents the number of clusters, n is the number of instances, and $E(C_i)$ denotes the individual entropy of the ith cluster and is given by :

$$E(C_i) = -\sum_{j=1}^{|\mathscr{C}_j|} \frac{|C_i \cap \mathscr{C}_j|}{|C_i|} log\left(\frac{|C_i \cap \mathscr{C}_j|}{|C_i|} \right),$$

where C_i are the clusters under consideration and $\mathscr{C}_i$ are the ground truth clusters (given by the data set documentation). In thermodynamics, entropy measures the amount of "disorder" of a system so, intuitively, a good clustering algorithm should make this index as low as possible.

The Adjusted Rand Index (*ARI*) is the corrected-for-chance version of the Rand index (Rand, 1971), which essentially evaluates the clustering result based on the relationship of pairwise instances. Let a be the number of pairs of instances that are placed in the same class C and in the same cluster $\mathscr{C}$, b be the number of pairs of instances that are in the same class C but not in the same cluster $\mathscr{C}$, c the number of pairs of instances that are in the same cluster $\mathscr{C}$ but not in the same class C, and d the number of pairs of objects that are in different classes and in different clusters in both partitions. Then *ARI* is defined as follows:

$$ARI = \frac{2(ad - bc)}{(a+b)(b+d) + (a+c)(c+d)},$$

5. Experiments

In this section we provide an experimental evaluation of our approach. The experiments are conducted on standard data sets and several metrics are employed in order to evaluate the performance of our algorithm (see Section 4.2.2). Furthermore, in order to better evaluate the results (especially the rough clusters), we perform an additional analysis (introduced in Section 4.2.1) for each case study. The purpose of this validation part is to analyze, for example, whether the instances assigned to the upper bounds may indeed represent hybrid data. We conduct three case studies on three well known datasets from UCI (Dua and Graff, 2017) namely Iris, Seeds and Wine and we validate the results using our proposed validation methodology from Section 4.2.1. For benchmarking reasons, we also conduct experiments on other datasets from UCI (Section 5.5) and we compute several evaluation measures that are needed in order to compare our approach with the related work.

5.1. Case study — Iris data set

The Iris data set (Fisher, 1936), one of the most widely used data sets in pattern recognition, contains 150 instances with four attributes denoting the sepal and petal length and width of three species of Iris plants. There are three classes with 50 instances each, one of the classes being linearly separable from the other two.

The data is scaled in the $[0, 1]$ range using Min-Max normalization and the algorithms described in Section 4.1 are applied with the following parameter setting: $i_{\max} = 100$, $\lambda = 100$, $\sigma_1 = 0.0115$, and $\sigma_2 = 0.1$.

In order to better understand the algorithm and the proposed validation methodology, we complement this case study with visual elements of the process. In Fig. 3, we show the official clusters of the Iris data set, as specified in the data set documentation. In order to graphically represent the data we have first performed a PCA analysis (S. and P., 1901) using Scikit-learn (Pedregosa et al., 2011) showing that the first two components capture over 97% of the total variation, and we used these two components for plotting the data.

As it may be seen in Fig. 3, only the first class (Setosa) is clearly separated from the other two classes. So it would not be surprising if a unsupervised learning algorithm would yield two or even more than three clusters. We would like to acknowledge that there are three classes, but at least the last two classes contain some peculiar instances that either seen more likely to belong to the other class or are quite far away from any class. *Our aim is to identify these hybrid instances while still accepting that there are three classes*. The main advantage of our approach is the fact that it offers more insight in the data by identifying possible outliers (e.g. possible errors in measurement) and rough instances, i.e., instances that have *some* common aspects with more than two classes, but cannot be accepted with certainty as belonging to either class (in this case possible hybrid species).

The first phase of the algorithm produces a great number of clusters that are shown is Fig. 4 with different colors. As it may be seen, there already are three clearly distinguishable large clusters. After the second phase of the algorithm the number of clusters is significantly reduced (Fig. 5), most of the instances belonging to three large clusters and the other ones forming very small groups. As a result of this phase some instances may belong to more than one cluster (rough instances), but they will only be marked with a distinct color after the last phase. The result of the final phase of the algorithm (without marking the hybrid instances) is shown in Fig. 6. In Fig. 7 we also mark the outliers and the rough instances identified as described in Section 4.

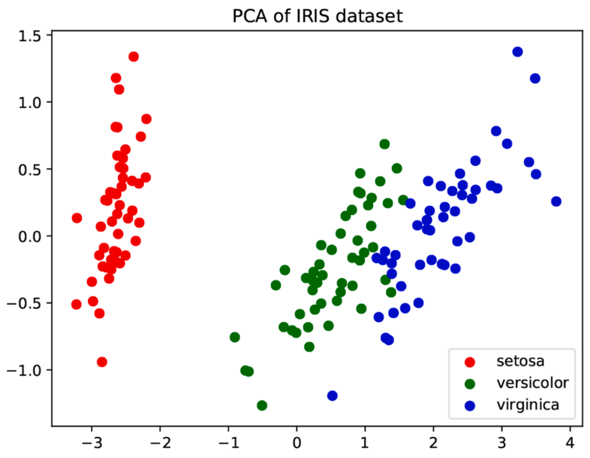

Fig. 3. Clusters in the Iris data set according to the official documentation.

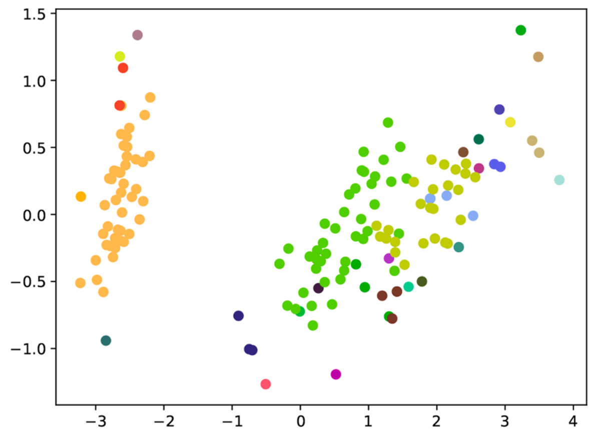

Fig. 4. Clusters in the Iris data set after the first phase of the algorithm.

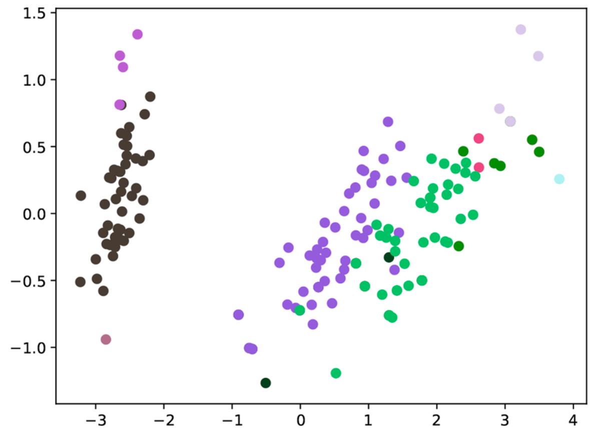

Fig. 5. Clusters in the Iris data set after the second phase of the algorithm.

5.1.1. Results and discussion

We obtain three clusters: C_1 with all instances from 1 to 50, C_2 mostly with instances from 51 to 100, and C_3 mostly with instances from 101 to 150. Instances 106, 107, 120, 123, 134, 135 are assigned to cluster C_2, but according to the data set documentation they should be in C_3. Also, instances 58, 60, 61, 69, 71, 73, 88, 94, 99 are assigned to cluster C_3, while the documentation specifies that they belong to cluster C_2. If we

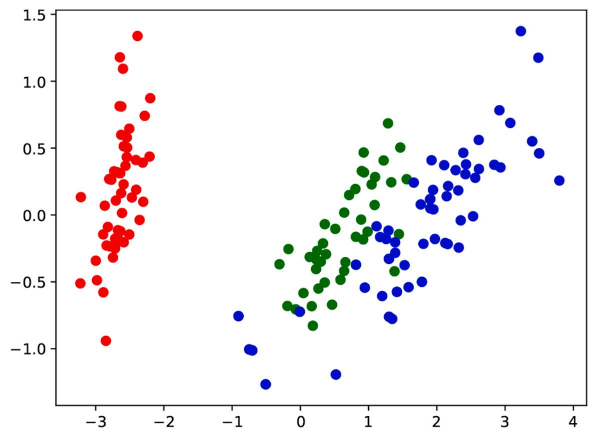

Fig. 6. Clusters in the Iris data after set the third phase of the algorithm without hybrid data.

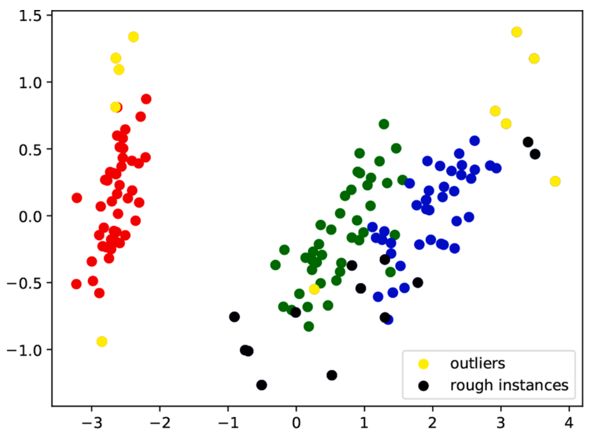

Fig. 7. Clusters in the Iris data set after the third phase of the algorithm with hybrid data.

consider all the aforementioned instances as classification errors then the accuracy is **90**%.

However, since we operate with rough clusters, we also obtain a finer grained information. The algorithm places the following instances in the upper approximation of clusters C_2 and C_3: 58, 60, 61, 69, 73, 88, 94, 99, 106, 107, 120, 123, 135. So these instances are actually assigned to the correct cluster, but because they are also close enough to another one, they are placed in both clusters, in their corresponding upper approximations. So these are not actually classification errors. The only errors (with respect to the data set documentation) are instances 71 and 134 which are not identified as rough instances by the algorithm and are placed in wrong clusters (lower approximation). This, arguably, leads to an accuracy of **98**.**66**%.

Furthermore, the following instances are marked as possible outliers: 15, 16, 33, 34, 42, 63, 110, 118, 119, 132, 136. The hybrid data (rough instances, outliers) identified in this case study, together with the classification errors (71 and 134) and the accuracy will be analyzed in the next sections in more detail.

5.1.2. *Validation*

In order to have a better insight of the results and in order to check if the instances that we report as belonging to upper approximations should indeed be considered as such, we conduct a deeper analysis as described in Section 4.2.1.

In Table 2 we show the cluster representatives together with the standard deviations within the data involved in computing the components of each representative. The contents of each cluster is exactly the one specified in the official data set documentation. So, for example, the representative of cluster C_1 (having the contents from the data set documentation) is $\langle 0.196, 0.595, 0.078, 0.061\rangle$.

In Table 3 we show the top 10 instances that are furthest away from the representative of each cluster.

In the next step, we eliminate the hybrid data identified by our algorithm from the original data set and we recompute the representatives and the list of instances that are most dissimilar within each cluster. If the hybrid data contains outliers then the representative computation could be affected so we would like to eliminate this situation as much as possible. After eliminating the hybrid instances from the data set, we recompute the cluster representatives (Table 4) and the list of most dissimilar instances (Table 5). As it may be seen from Tables 2, 3, 4, 5, the results are changed quite a bit and we decide to continue our analysis considering the results from Tables 4 and 5 which are based on the data set without the hybrid data.

In Table 6 we show the hybrid data obtained in our approach, as well as the classification errors (errors as per the Iris data set documentation). In columns C_1 C_2 C_3 we show the distances from each instance to the cluster representative (we use the cluster representatives from Table 4). For each cluster we specify the distance of the most dissimilar instance (see first row from Table 5). These distances represent the *rough thresholds*$\mathscr{T}_R^k$, corresponding to each cluster (see Section 4.2.1). The *outlier threshold*, $\mathscr{T}_O$, is also computed as described in Section 4.2.1 and, for the Iris data set, it is equal to 0.09.

In column *Rough*? we specify if an instance should have been detected by our algorithm as a rough instance and in column *Outlier*? we specify if it should have been detected as an outlier, as described in Steps 2 and 3 from Section 4.2.1. Finally, column *Wrong*? shows if the hybrid data identified by us should indeed fall under this category; the classification errors are also included in this analysis.

As we can see in Table 6, there are some rough instances which according to our analysis (4.2.1) should not have been identified as such. However, except for instance 60, all the other problematic rough instances are marked as outliers by our analysis. And hence the question mark in the column *Wrong*?. It is true that failing to identify the precise category (rough or outlier) for these instances is an inconvenience, but the fact that they are placed in either category (rough or outlier) is far more important, in our opinion.

But let us check the situation for instance 60 which was marked as rough by our algorithm and, according to the analysis, it is neither rough nor an outlier. Since the value of the outlier threshold, $\mathscr{T}_O$, is 0.09 for this data set (see Step 3 from Section 4.2.1) it is clear that instance 60 is not an outlier — the distance to cluster C_2 is only 0.062. In order to check if it should be rough we apply Step 2 from Section 4.2.1. The value of $\mathscr{T}_R^j$ is 0.099. The values of $\mathscr{T}_R^k$ are 0.073 for cluster C_1 and 0.091 for cluster C_3. But $0.486 - 0.062 > 0.073 + 0.099$ so instance 60 does belong to the upper approximation of cluster C_1. Also $0.279 - 0.062 > 0.091 + 0.099$ so it does not belong to the upper approximation of cluster C_3. Hence, according to our analysis, instance 60 is neither rough

Table 2
Cluster representatives and standard deviations in the Iris data set.

Attribute	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	μ	σ	μ	σ	μ	σ
1	0.196	0.097	0.454	0.142	0.636	0.175
2	0.595	0.156	0.321	0.129	0.406	0.133
3	0.078	0.029	0.553	0.079	0.772	0.093
4	0.061	0.043	0.511	0.082	0.802	0.113

Table 3
The most dissimilar instances within each cluster in the Iris data set.

Rank	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	Instance	*Distance*	*Instance*	*Distance*	*Instance*	*Distance*
1	42	0.242	61	0.206	107	0.309
2	16	0.205	58	0.152	132	0.272
3	34	0.123	94	0.151	118	0.257
4	15	0.108	51	0.126	120	0.187
5	33	0.085	99	0.121	119	0.185
6	14	0.078	53	0.108	110	0.145
7	9	0.077	78	0.094	123	0.139
8	19	0.063	86	0.084	136	0.117
9	39	0.061	71	0.08	106	0.112
10	6	0.056	57	0.078	135	0.111

Table 4
Cluster representatives and standard deviations in the Iris data set (without hybrid data).

Attribute	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	μ	σ	μ	σ	μ	σ
1	0.186	0.087	0.478	0.127	0.605	0.124
2	0.578	0.114	0.356	0.102	0.406	0.098
3	0.08	0.03	0.565	0.063	0.751	0.066
4	0.061	0.044	0.518	0.07	0.811	0.099

Table 5
The most dissimilar instances within each cluster in the Iris data set (without hybrid data).

Rank	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	Instance	*Distance*	*Instance*	*Distance*	*Instance*	*Distance*
1	19	0.073	51	0.099	114	0.091
2	14	0.068	82	0.089	131	0.088
3	9	0.067	53	0.083	108	0.085
4	6	0.066	54	0.078	130	0.079
5	17	0.065	81	0.075	122	0.073
6	39	0.052	78	0.074	126	0.069
7	45	0.04	86	0.064	101	0.066
8	43	0.032	80	0.06	115	0.066
9	13	0.032	57	0.056	145	0.06
10	47	0.032	87	0.054	109	0.057

nor an outlier, it is just a normal instance belonging to cluster C_2. So the fact that the algorithm has marked it as being rough is indeed an error. Fig. 8 shows instances 60 together with the cluster representatives.

As shown in Table 6, all outliers are correctly identified according to our analysis. However, we would like to have a deeper look at one of these outliers, specifically at instance 63. As it may be seen in Fig. 9, this instance seems to belong to cluster C_2. And indeed, according to Table 6, it is at a distance of 0.098 from cluster C_2, the distance of the most dissimilar instance in this cluster being 0.099. Since the value of the rough threshold, $\mathscr{T}_O$, is set to 0.09 for this data set (see Step 3 from Section 4.2.1), our validation methodology indicates that it is an outlier. This is unfortunately an error of the analysis (but is the only error of the validation methodology) and it is caused by the value of ϵ (see Section 4.2.1).

Instances 71 and 134 which should have been assigned, respectively, to clusters C_2 and C_3, according to the Iris data set documentation, are classified into wrong clusters by the algorithm: instance 71 is assigned to the lower approximation of cluster C_3 and instance 134 to the lower approximation of cluster C_2. The correct result, according to our analysis, would have been to identify these two instances as being rough (Table 6). The fact that we failed to identify this is an error according to our methodology. Nevertheless, we may observe that instance 71 is closest to cluster C_3, and instance 134 is closest to cluster C_2 (the clusters

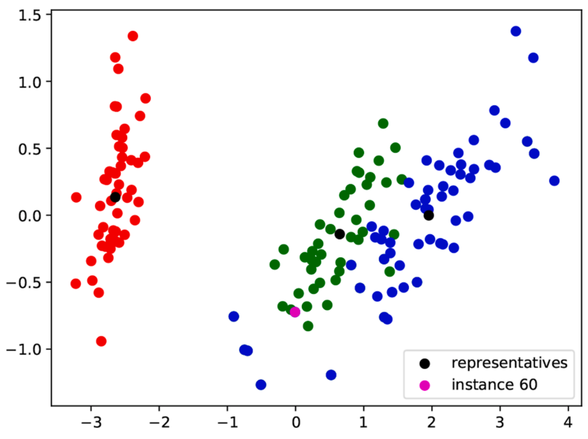

Fig. 8. Cluster representatives and instance 60.

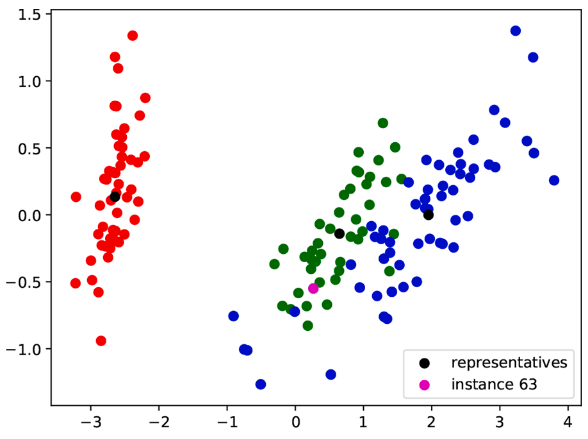

Fig. 9. Cluster representatives and instance 63.

Table 6
Hybrid data and classification errors in the Iris data set.

Instance		C_1 (0.073)	C_2 (0.099)	C_3 (0.091)	*Rough*?	*Outlier*?	*Wrong*?
Rough	58	0.364	0.184	0.57	false	**true**	**true?**
	60	0.486	0.062	0.279	false	false	**true**
	61	0.551	0.248	0.631	false	**true**	**true?**
	69	0.898	0.082	0.187	true	false	false
	73	0.884	0.041	0.101	true	false	false
	88	0.781	0.06	0.209	true	false	false
	94	0.4	0.185	0.568	false	**true**	**true?**
	99	0.331	0.149	0.511	false	**true**	**true?**
	106	1.912	0.443	0.137	false	**true**	**true?**
	107	0.767	0.142	0.277	true	true	false
	120	0.957	0.092	0.179	true	true	false
	123	1.954	0.453	0.167	false	**true**	**true?**
	135	0.927	0.058	0.109	true	false	false
Outliers	15	0.121	0.74	1.325	false	true	false
	16	0.224	0.808	1.314	false	true	false
	33	0.096	0.82	1.448	false	true	false
	34	0.137	0.809	1.393	false	true	false
	42	0.223	0.686	1.401	false	true	false
	63	0.609	0.098	0.371	false	true	false
	110	1.889	0.525	0.157	false	true	false
	118	2.053	0.661	0.284	false	true	false
	119	2.262	0.577	0.213	false	true	false
	132	1.924	0.625	0.302	false	true	false
	136	1.949	0.469	0.139	false	true	false
Err	71	0.81	0.064	**0.057**	**true**	false	**true?**
	134	0.847	**0.028**	0.063	**true**	false	**true?**

in which we assigned them). So, according to our analysis, not marking these instances as being rough is indeed an error, but they are after all assigned to the closest cluster. The question marks from the last column suggest that even though these two plants have strong similarities with two species, they seem to be more similar to the group in which we classified them.

5.2. Case study — Seeds

The Seeds data set (Kulczycki, 2012) contains 210 instances with seven attributes describing geometric parameters of wheat kernels. There are three classes of 70 instances each.

After scaling the data in the $[0,1]$ range using Min-Max normalization, the algorithms from Section 4.1 are applied with the following parameter setting: $i_{\max} = 100$, $\lambda = 100$, $\sigma_1 = 0.024$, and $\sigma_2 = 0.2$.

5.2.1. Results and discussion

We obtain three clusters: C_1 containing mostly instances between 1 and 70, C_2 with most instances between 71 and 140, and C_3 with instances from 141 to 210 in general. Instances 125, 166, 190, 193, 200, 202, 206 are assigned to cluster C_1, even though, according to the data set information, they do not belong in this group. Likewise, instances 9, 10, 13, 32, 37, 44 are assigned to cluster C_2, and instances 20, 30, 61, 64, 70 are assigned to cluster C_3. If we were to consider all these instances as true classification errors then the accuracy would be **91.42%**.

But as a result of our algorithm, the following instances are marked as being possibly rough: 13, 61, 82, 84, 91, 94, 95, 129, 130, 133, 138, 139, 190. Instances 13, 61 and 190 are assigned to the correct clusters (according to the documentation), but since they are also close to other ones, they are placed in the respective upper approximations as well. If we do not consider these three instances as classification errors then the accuracy becomes **92.85%**. We also obtain the following outliers: 61, 62, 78, 79, 83, 87, 88, 89, 90, 98, 99, 109, 112, 114, 115, 121, 165, 189, 204, 208.

5.2.2. Validation

In this section we would like to check if the rough instances and outliers identified in our approach should indeed be considered as such. We would also like to analyze the reported classification errors. This analysis is performed according to the steps described in Section 4.2.1.

In Table 7, we show the cluster representatives together with the standard deviations. The contents of each cluster is exactly to one from the original data set. In Table 8, we show the representatives and standard deviations after the outliers identified by us are eliminated. As observed from Tables 7 and 8, the standard deviations change significantly after outlier elimination, so we continue our analysis on the preprocessed data set. In Table 9 we show the top 10 most dissimilar instances from each cluster, the cluster contents being the one specified in the data set documentation.

In Table 10, we show the hybrid data obtained in our approach, as well as the classification errors (errors as per the Seeds data set documentation). In columns C_1, C_2 C_3 we show the distances from each instance to the cluster representative (we use the cluster representatives from Table 8). For each cluster we specify the distance of the most dissimilar instance (see first row from Table 9). These distances represent the *rough thresholds* $\mathscr{T}_R^k$, corresponding to each cluster. The *outlier threshold*, $\mathscr{T}_O$, is computed as described in Section 4.2.1 and, for the Seeds data set, it is equal to 0.2. Column *Wrong*? shows if an instance marked by us as an outlier, rough instance or as an error should indeed fall under this category. For brevity reasons, in Table 10, we only show the problematic rough instances and outliers — the ones that we marked as being rough or outliers but, according to this analysis, they aren't. However, we show all classification errors.

As we can see in Table 10, six of the instances marked by us as being rough should not have been marked as such. Out of these six instances, three (91, 95, 129) should have been marked as outliers. Marking them as being rough instead of outliers is an error, but we believe that the fact that their hybrid nature has been discovered is a good thing. Eight of the outliers reported by the algorithm are not outliers according to our analysis. Instance 165 is hybrid indeed, but it should have been marked as a rough instance. The other seven problematic outliers are indeed errors — as it may be seen in Table 10, they are quite close to some cluster centers (closer than the most dissimilar instance of that cluster). But let us observe that all these problematic outliers and rough instance have been assigned to the correct cluster.

All classification errors (as per the data set documentation) are shown in Table 10. Our analysis shows that nine of these instances (9, 10, 20, 37, 70, 125, 166, 200, 202) are in fact closest to the cluster assigned by the algorithm and hence they are not errors. As it may be seen, all these nine instances are also rough so they are close enough to more than one cluster, but they are closest to the cluster they were actually classified into, so maybe they really belong there.

Table 7
Cluster representatives and standard deviations in the Seeds data set.

Attribute	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	μ	σ	μ	σ	μ	σ
1	0.354	0.114	0.731	0.135	0.121	0.068
2	0.389	0.118	0.77	0.127	0.173	0.07
3	0.653	0.146	0.684	0.14	0.375	0.196
4	0.343	0.129	0.703	0.15	0.186	0.077
5	0.438	0.126	0.747	0.131	0.159	0.104
6	0.247	0.152	0.374	0.153	0.523	0.173
7	0.28	0.129	0.739	0.124	0.294	0.079

Table 8
Cluster representatives and standard deviations in the Seeds data set after outliers are eliminated.

Attribute	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	μ	σ	μ	σ	μ	σ
1	0.359	0.096	0.702	0.109	0.119	0.066
2	0.393	0.098	0.738	0.1	0.178	0.068
3	0.667	0.149	0.705	0.127	0.331	0.169
4	0.341	0.111	0.665	0.111	0.2	0.072
5	0.447	0.112	0.734	0.113	0.143	0.095
6	0.231	0.154	0.348	0.114	0.518	0.138
7	0.272	0.112	0.709	0.086	0.308	0.073

Table 9
The most dissimilar instances within each cluster in the Seeds data set.

Rank	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	Instance	*Distance*	*Instance*	*Distance*	*Instance*	*Distance*
1	60	0.381	136	0.352	142	0.227
2	38	0.34	135	0.21	198	0.213
3	17	0.321	123	0.204	196	0.211
4	40	0.31	120	0.181	180	0.17
5	24	0.246	134	0.158	197	0.149
6	63	0.242	81	0.146	159	0.144
7	52	0.225	101	0.135	171	0.14
8	19	0.199	76	0.135	150	0.137
9	5	0.167	72	0.107	175	0.134
10	26	0.167	113	0.105	143	0.133

5.3. Case study — Wine

The Wine data set (Forina, 1991) contains 178 instances with 13 attributes describing information of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.

After scaling the data in the $[0,1]$ range using Min-Max normalization, the algorithms from Section 4.1 are applied with the following

Table 10
Problematic hybrid data and classification errors in the Seeds data set.

Instance		C_1 (0.381)	C_2 (0.352)	C_3 (0.227)	*Rough?*	*Outlier?*	*Wrong?*
Rough	82	0.89	0.101	1.869	false	false	**true**
	84	1.241	0.176	2.483	false	false	**true**
	91	1.575	0.275	2.838	false	**true**	**true?**
	94	0.971	0.172	1.942	false	false	**true**
	95	1.467	0.369	2.092	false	**true**	**true?**
	129	1.382	0.218	2.641	false	**true**	**true?**
Outliers	79	1.256	0.154	2.111	false	false	**true**
	83	1.388	0.176	2.493	false	false	**true**
	87	0.872	0.085	2.098	false	false	**true**
	98	1.211	0.184	2.214	false	false	**true**
	99	0.886	0.096	1.702	false	false	**true**
	112	1.039	0.081	2.074	false	false	**true**
	165	0.684	2.022	0.092	**true**	false	**true?**
	189	0.799	2.09	0.178	false	false	**true**
Errors	9	0.383	**0.094**	1.145	**true**	false	**true?**
	10	0.212	**0.151**	1.035	**true**	false	**true?**
	20	0.156	1.217	**0.103**	**true**	false	**true?**
	30	0.084	0.865	0.143	true	false	**true**
	32	0.08	0.323	0.562	true	false	**true**
	37	0.18	**0.156**	0.814	**true**	false	**true?**
	44	0.207	0.233	0.664	true	true	**true**
	64	0.103	0.946	0.118	true	false	**true**
	70	0.234	1.236	**0.053**	**true**	false	**true?**
	125	**0.16**	0.45	0.904	**true**	false	**true?**
	166	**0.208**	1.447	0.226	**true**	true	**true?**
	193	0.239	1.421	0.138	true	false	**true**
	200	**0.159**	1.312	0.38	**true**	false	**true?**
	202	**0.21**	1.47	0.449	**true**	true	**true?**
	206	0.208	1.443	0.15	true	false	**true**

parameter setting: $i_{\max} = 100$, $\lambda = 100$, $\sigma_1 = 0.145$, and $\sigma_2 = 0.45$.

5.3.1. *Results and discussion*

We obtain three clusters: C_1 is composed mostly of instances from 1 to 59, C_2 with instances between 60 and 130 in most cases, and C_3 with most instances greater than 130. Instance 74 is assigned to cluster C_1, but according to the documentation it belongs to the second cluster. Instance 26 is assigned to cluster C_2 instead of being classified into the first one. Cluster C_3 contains the following instances that belong to other clusters according to the Wine data set: 71, 78, 84, 106, 113, 119, 123, 124, 128, 130. This leads to an accuracy of **93.25%**.

Our algorithm marks the following instances as being rough: 61, 78, 94, 106, 113, 119, 123, 124, 128, 130. Instances 78, 106, 113, 119, 123, 124, 128, and 130 are assigned to more than one cluster, including the officially correct one so we do not consider them classification errors. Then then accuracy becomes **97.75%**. The following instances are marked as outliers: 26, 60, 61, 62, 69, 70, 74, 76, 79, 84, 96, 97, 111, 116, 122, 125, 137, 138, 147, 159, 160.

5.3.2. *Validation*

In this section we would like to check if the rough instances and outliers identified in our approach should indeed be considered as such. We would also like to analyze the reported classification errors. This validation step is performed as descried in Section 4.2.1.

In Table 11, we show the cluster representatives together with the standard deviations. The contents of each cluster is exactly to one from the original data set. In Table 12, we show the representatives and standard deviations after the outliers and rough instances identified by us are eliminated. We continue the analysis on this preprocessed data set. In Table 13 we show the top 10 most dissimilar instances from each cluster, the cluster contents being the one specified in the data set documentation.

In Table 14, we show the hybrid data obtained in our approach, as well as the classification errors (errors as per the Wine data set documentation). In columns C_1, C_2 C_3 we show the distances from each instance to the cluster representative (we use the cluster representatives from Table 12). For each cluster we specify the distance of the most

Table 11
Cluster representatives and standard deviations in the Wine data set.

Attribute	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	μ	σ	μ	σ	μ	σ
1	0.714	0.121	0.329	0.141	0.559	0.138
2	0.251	0.135	0.236	0.199	0.513	0.213
3	0.586	0.12	0.473	0.168	0.576	0.098
4	0.332	0.13	0.497	0.171	0.558	0.115
5	0.395	0.113	0.267	0.181	0.319	0.117
6	0.641	0.116	0.441	0.187	0.241	0.122
7	0.557	0.083	0.367	0.148	0.093	0.061
8	0.302	0.131	0.441	0.232	0.599	0.232
9	0.47	0.129	0.385	0.189	0.235	0.128
10	0.362	0.105	0.154	0.078	0.522	0.195
11	0.473	0.094	0.469	0.164	0.165	0.092
12	0.692	0.13	0.555	0.181	0.151	0.099
13	0.598	0.157	0.172	0.111	0.251	0.081

Table 12
Cluster representatives and standard deviations in the Wine data set after outliers are eliminated.

Attribute	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	μ	σ	μ	σ	μ	σ
1	0.718	0.119	0.335	0.138	0.555	0.127
2	0.251	0.136	0.178	0.127	0.504	0.192
3	0.579	0.109	0.463	0.137	0.569	0.096
4	0.325	0.119	0.499	0.126	0.55	0.114
5	0.392	0.111	0.203	0.081	0.33	0.116
6	0.643	0.116	0.459	0.182	0.232	0.095
7	0.559	0.083	0.385	0.117	0.094	0.059
8	0.296	0.124	0.403	0.183	0.583	0.235
9	0.47	0.13	0.379	0.126	0.221	0.093
10	0.365	0.103	0.151	0.073	0.519	0.173
11	0.472	0.094	0.485	0.149	0.167	0.092
12	0.691	0.131	0.594	0.135	0.155	0.098
13	0.601	0.156	0.158	0.094	0.255	0.081

Table 13
The most dissimilar instances within each cluster in the Wine data set.

Rank	*Cluster 1*		*Cluster 2*		*Cluster 3*	
	Instance	*Distance*	*Instance*	*Distance*	*Instance*	*Distance*
1	14	0.413	72	0.493	131	0.543
2	40	0.402	93	0.446	135	0.41
3	15	0.386	67	0.445	153	0.407
4	34	0.355	100	0.44	134	0.362
5	19	0.355	75	0.428	151	0.356
6	44	0.35	99	0.392	152	0.336
7	51	0.347	80	0.346	170	0.312
8	22	0.326	64	0.345	158	0.308
9	4	0.321	65	0.325	156	0.295
10	42	0.284	63	0.323	154	0.28

dissimilar instance (see first row from Table 13) These distances represent the *rough thresholds* $\mathcal{T}_R^k$, corresponding to each cluster. The *outlier threshold*, $\mathcal{T}_O$, is computed as described in Section 4.2.1 and, for the Wine data set, it is equal to 0.45. Column *Wrong*? shows if an instance marked by us as an outlier, rough instance or as an error should indeed fall under this category. For the sake of brevity, in Table 14, we only show the problematic (according to this analysis) rough instances and outliers. However, we show all classification errors.

All ten instances marked by the algorithm as being rough are correct according to this analysis and only five out of the 21 outliers are problematic (see Table 14). But out of these five problematic outliers, the first four are in fact rough instances. So even though the precise nature of these hybrid items was not correctly identified, the fact that they are somehow marked is a positive aspect. On the other hand, instance 147 is neither an outlier nor rough, it is a normal instance belonging to cluster C_3. So marking it as an outlier is indeed an error.

In Table 14 we also show all classification errors according to the Wine data set documentation. All these instances should be rough according to our analysis so they are close enough to at least two clusters. Instances 71, 74, and 84 are closest to the cluster they were classified into so based on our analysis they actually belong there. Then the only true error is instance 26.

5.4. Case studies summary

In order to evaluate the quality of our approach, we use several cluster evaluation measures presented in Section 4.2.2. For each data set, we execute the algorithm 30 times and we present the average value for each evaluation measure in Table 15. We would like to mention that the number of executions was chosen to be 30 only because we would like to be consistent with other approaches we compare with. In line *Official* from Table 15 we show the index values for the clusters given in the official data set documentation. In line *ABARC* we show the average index values for the clusters obtained by our algorithm with all hybrid data included. Line *ABARC* − *O* shows the average index values for our clusters after eliminating the outliers.

As it may be seen in Table 15, the cluster structure proposed in the official documentation outperforms the results proposed by us only in terms of accuracy, ARI and entropy (of course). The best results for each index is marked in bold face. Even without eliminating the outliers, all results (except for the accuracy, ARI and entropy) are better compared to the official ones, but after eliminating the outliers the results are significantly improved in some cases.

In Table 16, we show the most frequently identified rough instances in 50 executions of the algorithm. For each data set, on the first line, we show the instance number and on the second line the occurrence rate (expressed in %).

5.5. Experiments on UCI datasets and performance evaluation

In order to evaluate our approach in comparison to other approaches, we consider several datasets from UCI (Dua and Graff, 2017) as shown in Table 17. We execute the algorithm 30 times for each dataset and we compute the mean values for several metrics. In Table 18 and in Table 19 we show the results before and after outlier elimination respectively. As it may be seen, after outlier elimination the results are greatly improved in some cases.

As we may see from Table 18 and from Table 19, the worst results are achieved for the Ecoli and Yeast datasets and therefore we wanted to investigate this situation in more detail. We decided to perform a Principal Component Analysis (PCA) and to graphically represent these datasets based on the principal components in order to have a better image of their internal structure. As we can see from Fig. 10, the Ecoli

Table 14
Hybrid data and classification errors in the Wine data set.

Instance		C_1 (0.413)	C_2 (0.493)	C_3 (0.543)	*Rough*?	*Outlier*?	*Wrong*?
Outliers	62	1.286	0.615	0.444	**true**	false	**true?**
	76	1.156	0.308	0.776	**true**	false	**true?**
	84	1.344	0.732	0.292	**true**	false	**true?**
	137	2.038	1.274	0.326	**true**	false	**true?**
	147	2.254	1.608	0.402	false	false	**true**
Errors	26	0.624	0.697	1.246	true	true	**true**
	71	1.116	0.505	**0.441**	**true**	false	**true?**
	74	**0.785**	1.044	2.146	**true**	true	**true?**
	84	1.344	0.732	**0.292**	**true**	false	**true?**

Table 15
Cluster quality measures.

		DB↓	*DN*↑	*SI*↑	*Entropy*↓	*ARI*↑	*Accuracy*(%)↑
Iris	*Official*	0.511	2.484	0.624	**0**	**1**	**100**
	ABARC	0.464	3.012	0.641	0.116	0.77	95.556
	ABARC − *O*	**0.354**	**3.616**	**0.716**	0.067	0.92	97.56
Seeds	*Official*	0.527	2.699	0.561	**0**	**1**	**100**
	ABARC	0.463	3.447	0.614	0.293	0.51	90.952
	ABARC − *O*	**0.21**	**5.972**	**0.772**	0.073	0.6	98.121
Wine	*Official*	0.98	1.474	0.465	**0**	**1**	**100**
	ABARC	0.968	1.6	0.48	0.11	0.65	96.985
	ABARC − *O*	**0.425**	**3.804**	**0.677**	**0**	0.85	**100**

Table 16
Rough instances in the Iris, Seeds and Wine data sets.

Iris	I#	60	135	107	99	150	94	58	138	104	147
———	%	98	80	40	38	38	38	38	34	34	34
Seeds	I#	133	193	206	60	190	40	189	13	101	105
	%	86	62	62	60	56	48	46	44	40	38
Wine	**I#**	77	94	177	8	47	44	42	95	101	64
	%	66	32	32	30	30	24	22	22	20	20

Table 17
Datasets utilised for benchmarking.

Dataset	Clusters	Instances	Attributes
Iris	3	150	4
Wine	3	178	13
Seeds	3	21	7
Ecoli	8	336	7
Cancer	2	683	9
Yeast	10	1484	8
Segmentation	7	2100	19
Page-Blocks	5	5473	10

Table 18
Results before eliminating outliers.

Dataset	DB ↓	DU ↑	SI ↑	Acc ↑	Entropy ↓	ARI ↑
Iris	0.46	3.01	0.64	95.55	0.11	0.77
Wine	0.96	1.6	0.48	96.98	0.11	0.65
Seeds	0.46	3.44	0.61	90.95	0.29	0.51
Ecoli	16.002	0.041	0.003	65.774	0.13	0.44
Cancer	0.607	2.107	0.645	95.608	0.077	0.836
Yeast	21.901	0.075	0.055	41.341	0.423	0.11
Segmentation	3.813	0.204	0.302	64.358	0.122	0.548
Page-Blocks	3.651	0.264	0.0	96.273	0.003	0.978

Table 19
Results after eliminating outliers.

Dataset	DB ↓	DU ↑	SI ↑	Acc ↑	Entropy ↓	ARI ↑
Iris	0.35	3.61	0.71	97.56	0.06	0.92
Wine	0.42	3.8	0.67	100	0	0.85
Seeds	0.21	5.97	0.77	98.12	0.07	0.6
Ecoli	13.378	0.061	0.04	68.439	0.11	0.548
Cancer	0.545	2.208	0.782	97.137	0.053	0.94
Yeast	9.906	0.159	-0.056	45.437	0.349	0.46
Segmentation	3.813	0.204	0.302	64.358	0.122	0.548
Page-Blocks	2.715	0.193	0.0	96.922	0.001	0.982

dataset contains many hybrid data items given by outliers, rough instances and overlapping clusters. The presence of hybrid data alone makes the classification process challenging. Moreover, this dataset is highly imbalanced: there are 336 instances grouped in 8 clusters, but three of these clusters are very small (the imL and imS classes contain only 2 instances, while the omL class contains only 5 instances). In the PCA analysis, we have also checked the explained variance ratio of each principal component and we have obtained the following values for the first and second component respectively: 0.3151 and 0.209. So even though these components explain just a little over 50% of the overall variability, from the graphical representation of the dataset we can still get some idea of the way the data is structured as well as some possible challenges to deal with, e.g., outliers, rough instances, overlapping clusters. A similar attempt was made for the Yeast dataset. Unfortunately, the explained variance ratios in this case were 0.2267 and 0.1588 for the first and second component respectively, so only about 36% of the total variance. For this reason we did not include a graphical representation in this case, but we suppose that this dataset presents similar issues with the Ecoli dataset, probably at a larger scale and we plan to

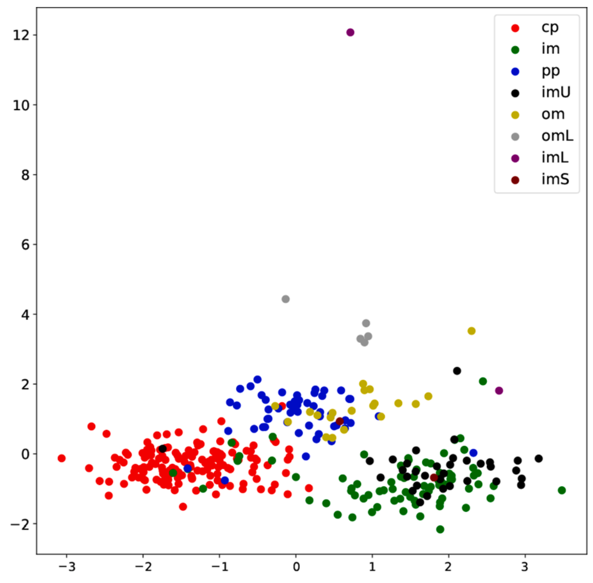

Fig. 10. Clusters in the ecoli dataset.

investigate this in the future.

In order to evaluate the scalability of our approach, we have measured the execution time on four synthetic data sets with 100, 1000, 10000, and 100000 instances respectively. To this end, the Benchee benchmaring libary (School, 2012) was used on a quad core *i7* laptop. Table 20 shows the average execution times in number of seconds for each data set, as well as the number of iterations per second. The last line of the table shows the decrease factor of the number of iterations per second (*ips*) with respect to the first data set (100 instances): so with 1000 instances the algorithm is 3.52 times slower (in terms of *ips*) than when executing it with 100 instances; also, with 10000 instances the algorithm is only 41.1 times slower (in terms of *ips*) than when running it with 100 instances.

As it may be seen in Table 20, as well as in Fig. 11, the decrease in performance is sub-linear with respect to the number of instances. So, if the number of instances is increased 10 times, the performance decreases only 3.52 times. Also, if the number of instances is increased 100 times, the performance decreases 41.1 times (in terms of *ips*).

6. Comparison to related work

This section presents a comparison with other classification algorithms in terms of accuracy and some widely used cluster evaluation measures. We would like to point out that, in our opinion, an *objective* and *relevant* validation is the one from Section 5.4 which is based on our

Table 20
Average execution times.

Number of instances	100	1000	10,000	100,000
Time (s)	0.03	0.13	1.56	15.51
Iterations per second	26.29	7.46	0.64	0.0645
Decrease factor	—	3.52	41.1	407.75

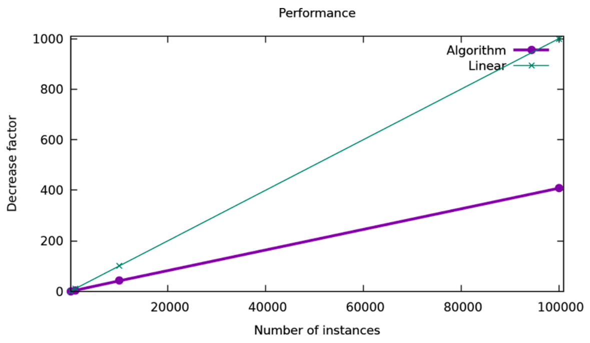

Fig. 11. Execution performance.

validation methodology proposed in Section 4.2, taking into account the official data set information. One of the motivations behind the validation methodology is the lack of official information regarding hybrid data, i.e., the difficulty of validating these results, which represent the added value of our approach. Another argument is the fact that cluster quality indices may be biased, as discussed in 4.2.2. Nevertheless, for benchmarking purposes, in this section we compare the performance of our clustering algorithm with other classification methods by trying to reproduce the same experimental setup as in the related works. In this sense, we execute our algorithm 30 times for each data set and we record the average, best, worst, and median values for the accuracy and entropy in order to compare them with several other clustering algorithms: GWOTS (Aljarah et al., 2019), GWO (Mirjalili et al., 2014), TS (F, 1989), DE (Storn and Price, 1997), PSO (Kennedy, 2010), GA (Goldberg, 1989), SA (Osman and Christofides, 1994). We also present comparative results in terms of accuracy, *DB* index, and *DN* index with other classification approaches including some fuzzy and rough and ones (Aeberhard et al., 1994; Charytanowicz et al., 2010; Maji and Pal, 2012a; Patel et al., 2014; Scott et al., 2003). For benchmarking purposes, we have extended our experiments on several datasets from Dua and Graff (2017), as shown in Table 18 and in Table 19 and we compare our results with several other clustering algorithms (Bryant and Cios, 2017; Khan et al., 2014; Liang et al., 2022; Rodriguez and Laio, 2014; Viswanath and Pinkesh, 2006) in terms of Accuracy (Table 26) and Adjusted Rand Index (Table 27).

The comparison of our algorithm (ABARC) with other clustering approaches in terms of accuracy (average, median, best, worst) is shown in Table 21 for the Iris, Seeds, and Wine data sets. We have denoted with ABARC–O the results of our algorithm after outlier elimination. As it may be seen in Table 21, our approach outperforms all the other algorithms in terms of accuracy in all cases (average, median, best, worst), even without outlier elimination. By eliminating the outliers, the results become far better in some cases.

In Table 22, we show comparative results of our approach with other clustering algorithms in terms of entropy (average, median, best, worst) obtained after 30 executions. As it may be seen, our algorithm outperforms all the other algorithms in terms of entropy in all cases (average, median, best, worst), on the considered data sets, even without outlier elimination. By excluding the outliers all results are better, greatly improved in some cases.

In Table 23 we show comparative results in terms of accuracy by including the best results that we know of on the considered data sets. We would like to outline that some of the methods we compare with in Table 23 are in fact supervised learning approaches, so such a comparison is not quite in our advantage. For our algorithm we report the average accuracy values after 30 executions, with and without outliers, as in Table 15 from Section 5.4.

In Scott et al. (2003), the authors apply the Fuzzy c-Means algorithm and Principal Component Analysis (PCA) (S. and P., 1901) on the Iris data set reporting an accuracy of 88.6%. They show that clusters C_2 and C_3 overlap and they suggest the presence of outliers. Unfortunately, neither the instances that may belong to more than one cluster nor the outliers are given. As shown in Table 23, our algorithm clearly outperforms the approach from Scott et al. (2003). In Patel et al. (2014), a comparison of various classifiers is performed on the Iris data set. Using 10-fold cross validation, the best result (an accuracy of 97.3%) is obtained for the Multilayer Perceptron implementation from Weka (Hall et al., 2009). However, it is important to note that this result is obtained in a *supervised learning* context. Even so, the *clustering* algorithm proposed by us performs better after outlier elimination, as shown in Table 23. For the Seeds data set, the best result in terms of accuracy that

Table 21
Accuracy (average, best, worst, median) clustering results.

	Algorithm	Average	Median	Best	Worst
Iris	GWOTS	88.667	88.667	88.667	88.667
	GWO	82.2	88.667	88.667	66.667
	TS	82.067	88.667	88.667	66.667
	DE	86.733	87.333	93.333	82
	PSO	77.133	77.333	88.667	66.667
	GA	75.333	74	92.667	66.667
	SA	75.067	71.667	90	66.667
	ABARC	95.556	96.667	96.667	88.0
	ABARC–O	**97.56**	**98.182**	**99.074**	**89.216**
Seeds	GWOTS	88.857	89.048	89.048	88.571
	GWO	87.048	90	91.429	65.714
	TS	86.524	88.810	89.048	65.714
	DE	77.048	80.476	85.238	62.857
	PSO	76.095	75.000	88.571	62.381
	GA	76.762	77.143	86.667	64.762
	SA	71.476	70	87.619	61.429
	ABARC	90.952	90.952	90.952	90.952
	ABARC–O	**98.121**	**98.095**	**98.98**	**98.058**
Wine	GWOTS	95.506	95.506	96.629	93.820
	GWO	59.888	62.360	64.607	39.888
	TS	92.079	94.944	96.629	62.921
	DE	58.202	60.955	73.034	39.888
	PSO	41.966	39.888	60.112	39.888
	GA	59.888	61.236	69.663	44.944
	SA	52.584	58.427	62.921	39.888
	ABARC	96.985	97.191	97.191	95.506
	ABARC–O	**100**	**100**	**100**	**100**

Table 22
Entropy (average, best, worst, median) clustering results.

	Algorithm	Average	Median	Best	Worst
Iris	GWOTS	0.26358	0.26358	0.26358	0.26358
	GWO	0.32045	0.26358	0.26358	0.46659
	TS	0.31529	0.26358	0.26358	0.46659
	DE	0.27222	0.27664	0.21191	0.34021
	PSO	0.34250	0.34402	0.24642	0.42062
	GA	0.39998	0.40526	0.19146	0.51821
	SA	0.36360	0.40394	0.24595	0.42062
	ABARC	0.116	0.115	0.073	0.204
	ABARC–O	**0.067**	**0.074**	**0.031**	**0.076**
Seeds	GWOTS	0.33019	0.32660	0.32600	0.33557
	GWO	0.43362	0.31305	0.27930	0.55760
	TS	0.35329	0.33108	0.32660	0.55760
	DE	0.44985	0.41566	0.33300	0.63013
	PSO	0.45737	0.48304	0.32839	0.59974
	GA	0.45985	0.46180	0.36536	0.57000
	SA	0.52086	0.54009	0.33188	0.68006
	ABARC	0.293	0.293	0.293	0.293
	ABARC–O	**0.073**	**0.074**	**0.045**	**0.076**
Wine	GWOTS	0.14427	0.14471	0.11713	0.18701
	GWO	0.61111	0.57427	0.57743	0.97907
	TS	0.18998	0.15222	0.11713	0.58062
	DE	0.69294	0.59598	0.51678	0.97426
	PSO	0.94772	0.98855	0.60128	0.98855
	GA	0.72193	0.73591	0.57990	0.92031
	SA	0.76438	0,67757	0.54954	0.98855
	ABARC	0.11	0.104	0.103	0.145
	ABARC–O	**0**	**0**	**0**	**0**

Table 23
Comparison to related work in terms of accuracy (Iris, Wine, Seeds).

Algorithm	*Dataset*	*Acc*(%)
PCA+FCM (Scott et al., 2003)	Iris	88.6
MLP (Patel et al., 2014)	Iris	97.3
ABARC	Iris	95.556
ABARC–O	Iris	**97.56**
CGCA (Charytanowicz et al., 2010)	Seeds	91.9
ABARC	Seeds	90.952
ABARC–O	Seeds	**98.121**
RDA (Aeberhard et al., 1994)	Wine	**100**
ABARC	Wine	96.985
ABARC–O	Wine	**100**

we know of (91.9%) is reported in Charytanowicz et al. (2010) where the proposed clustering algorithm is compared with k-Means outperforming it. As seen in Table 23, with an accuracy of 90.952%, our approach is quite close to the CGCA algorithm (Charytanowicz et al., 2010), outperforming it after outlier elimination with an accuracy of 98.121%. The authors from Aeberhard et al. (1994) compare different pattern recognition methods using the Wine data set among other standard and synthetic ones. They use the leave one out cross validation technique in all experiments and some of the classifiers are applied on a reduced feature space obtaining accuracies ranging from 74.1% to even 100%. We would like to point out again that these results are obtained in a *supervised learning context*. As shown in Table 15, the accuracy of our algorithm is 96.985% on this data set and we also obtain 100% accuracy after eliminating the outliers.

We also compare our algorithm with other clustering algorithms based on rough sets (Maji and Pal, 2012a) in terms of some widely used indexes. But, as argued in Section 4.2.1, such a comparison may be biased if the way these algorithms work is, essentially, by improving these indexes, which is the case for all the algorithms from Maji and Pal (2012a). In our opinion this is again not in our advantage, but we provide the comparison nonetheless. Table 24 shows the comparison (*DB* index, *Dunn* index) on Iris data set, while Table 25 shows the comparison (*DB* index, *Dunn* index) on Wine data set.

In Maji and Pal (2012a), several fuzzy and rough hybridizations of the k-Means algorithm are presented and compared. On the Iris data set, the reported DB index values range from 0.21 to 0.48 and the Dunn index values range from 2.63 to 7.75. On the Wine data set, the reported DB index values range from 0.19 to 0.968 and the Dunn index values range from 1.4 to 3.87. As seen in Table 24 and in Table 25, the index values obtained by us are not the best. In fact, the *DB* index that we obtain on the Wine data set is the worst one, but it improves after outlier elimination. The Dunn index on the same data set is better though. In fact, it is the second best on this data set after outlier elimination. Unfortunately, we don't know what distance measure is utilized in Maji and Pal (2012a) for computing the indexes and we also don't know exactly which instances are rough in order to have a more detailed comparison.

Table 24
Comparison to related work in terms of cluster quality on Iris data set.

Algorithm	*DB*↓	*DN*↑
FCM	0.32	4.32
FPCM	0.48	2.63
FPCM^{MR}	0.35	4.36
KHCM	0.32	4.27
KFCM	0.29	4.89
KFPCM	0.46	3.22
RCM	0.22	6.75
RFCM^{MBP}	0.22	6.91
RFCM	0.22	6.94
RPCM	0.22	7.13
RFPCM	**0.21**	**7.75**
ABARC	0.464	3.012
ABARC–O	0.354	3.616

Table 25
Comparison to related work in terms of cluster quality on Wine data set.

Algorithm	*DB*↓	*DN*↑
HCM	0.20	2.52
FCM	**0.19**	2.78
FPCM	0.45	1.40
FPCM^{MR}	0.88	1.78
KHCM	0.20	2.52
KFCM	**0.19**	2.76
KFPCM	0.48	1.54
RCM	0.20	3.08
RFCM^{MBP}	**0.19**	3.19
RFCM	**0.19**	3.13
RPCM	0.20	3.09
RFPCM	**0.19**	**3.87**
ABARC	0.968	1.6
ABARC–O	0.425	3.804

We also do not know if these results are the mean values obtained after several executions of these algorithms (like in our case). Moreover, since we deal with algorithms based on k-Means, they naturally tend to improve these indexes making the results biased (as argued in Section 4.2.2). *This is another reason for which we proposed what we believe to be an objective methodology for assessing the quality of a clustering solution when dealing with rough instances and outliers (Section 4.2).*

In Liang et al. (2022), a very interesting biology inspired clustering algorithm is introduced. It simulates the division and aggregation of cells and it is uses a validation and correction mechanism in order to continuously evaluate the cohesion of cell groups. The algorithm consists of two phases. In the first phase, the input data is divided into several cohesive groups, while in the second phase the groups are merged in a density-reachable manner. The authors compare their approach in terms of Accuracy and Adjusted Rand Index with other density-based clustering approaches based on the classical DBSCAN algorithm (Khan et al., 2014) like Fast-DBSCAN, DensityPeak, RNN-DBSCAN (Bryant and Cios, 2017; Rodriguez and Laio, 2014; Viswanath and Pinkesh, 2006). In Table 26 we show comparative results between our approach and the aforementioned ones in terms of Accuracy, while in Table 27 we show comparative results in terms of Adjusted Rand Index (ARI). From Table 26, it may be seen that for the Ecoli and Yeast datasets our algorithm is outperformed by all the other ones in terms of Accuracy. On the other hand, in terms of ARI we are on the third position, quite close to the second best (0.548 versus 0.55) in case of Ecoli and on the second last position in case of Yeast. Considering the Image Segmentation dataset we notice that we outperform only two of the other approaches in terms of Accuracy, but we outperform all of them in terms of ARI. For all other datasets we outperform the other algorithms in both Accuracy and ARI, and in some cases, especially after eliminating the outliers, we outperform the other algorithms by a significant margin. In order to better visualize the results in comparison with the related work, in Fig. 12 and in Fig. 13, we show the performance of our approach and the average performance of the related work in terms of accuracy and ARI, respectively. As Fig. 12 and in Fig. 13 illustrate, our approach significantly outperforms the related work in most cases and is outperformed only in a few cases and only by a small margin.

For verifying from a statistical viewpoint that this difference is statistically significant, a two-sided paired Wilcoxon signed-rank test was applied (Siegel and Castellan, 1988). The ARI values obtained for all evaluations of our algorithm from Table 27 were tested against the sample of average values obtained for the related work. A *p*-value of 0.046875 was obtained, showing that the performance of our approach differs significantly from the performance of the related work approaches, at a significance level of $\alpha = 0.05$. Thus, we may conclude that the improvement brought by our algorithm compared to existing related work is statistically significant, at a confidence level of 95%. A similar procedure applied for accuracy (seen in Table 26) produced a *p*-value

Table 26
Comparison to related work in terms of accuracy.

Dataset	SIGG	DBSCAN	FAST-DBSCAN	DensityPeak	RNN-DBSCAN	ABARC	ABARC-O
Iris	0.823	0.741	0.744	0.811	0.798	0.955	**0.975**
Ecoli	**0.831**	0.791	0.812	0.819	0.763	0.657	0.684
Wine	0.878	0.742	0.779	0.832	0.828	0.971	**1**
Breast Cancer	0.793	0.8	0.702	0.791	0.817	0.956	**0.971**
Yeast	**0.867**	0.796	0.689	0.695	0.68	0.413	0.454
Image Segmentation	0.761	0.672	**0.847**	0.733	0.821	0.643	0.76
Page-Blocks	0.826	0.711	0.598	0.880	0.636	0.962	**0.969**

Table 27
Comparison to related work in terms of ARI.

Dataset	SIGG	DBSCAN	FAST-DBSCAN	DensityPeak	RNN-DBSCAN	ABARC	ABARC-O
Iris	0.712	0.67	0.347	0.392	0.574	0.779	**0.92**
Ecoli	0.55	**0.631**	0.515	0.366	0.451	0.44	0.548
Wine	0.539	0.442	0.215	0.461	0.402	0.656	**0.854**
Breast Cancer	0.631	0.372	0.416	0.485	0.525	0.836	**0.94**
Yeast	**0.708**	0.548	0.472	0.433	0.56	0.11	0.46
Image Segmentation	0.411	0.277	0.379	0.584	0.423	0.548	**0.796**
Page-Blocks	0.479	0.551	0.443	0.608	0.487	0.978	**0.982**

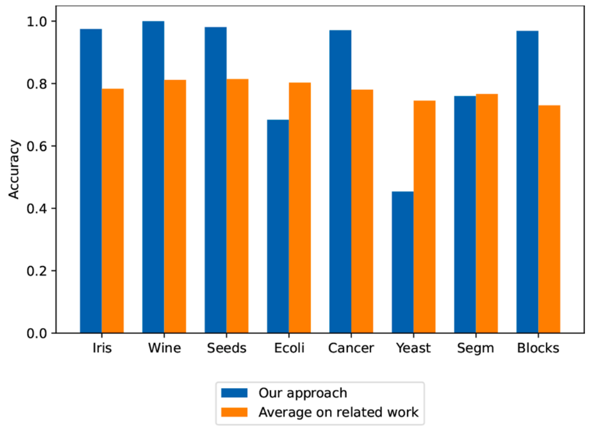

Fig. 12. Comparison to related work in terms of accuracy.

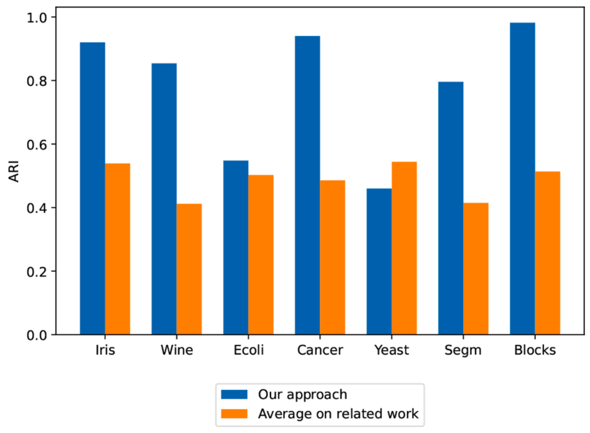

Fig. 13. Comparison to related work in terms of ARI.

greater than 0.05, so the null hypothesis could not be rejected in this case. Indeed, by examining the data from Fig. 12, even though we outperform the related work in most cases, the differences are not very high so perhaps more tests on other datasets should be considered in the future. On the other hand, as we have tried to explain throughout the paper, the focus and the added value of our approach is the discovery of hybrid data. We also propose a methodology for evaluating such an algorithm in Section 4.2 and we argue about the drawbacks of various evaluation measures. One such evaluation measure is accuracy which, first and foremost, is not an ideal way to evaluate an *unsupervised* learning approach, but we use it because we need a way to compare with other approaches, both supervised and unsupervised ones. In conclusion, even though in most cases we obtain better results in terms of accuracy compared to the related work, we cannot say that the performance or our approach differs significantly.

7. Conclusions and future work

In this paper we have introduced an agglomerative clustering algorithm which uses notions from rough sets theory in order to discover hybrid data. By hybrid data we mean outliers or rough instances, i.e., instances that share common traits with more than one cluster. By identifying hybrid instances, our clustering algorithm offers more insight to the analyzed data compared to other approaches. By identifying outliers (which may be caused by measurement errors), the data analyst has the possibility of eliminating them which could have a significant impact on the classification process in some cases, as it may be seen in Section 5.4. Identifying rough instances could also provide valuable information to the data analyst: they could suggest the presence of a new class, or, in the Iris data set case, the possible existence of mutant individuals. Due to the use of software agents which execute in parallel, the algorithm is scalable. It is thus able to execute within seconds on an ordinary personal computer, with data sets containing hundreds of thousands of instances. Furthermore, we introduced a methodology for assessing the quality of the discovered hybrid data. The introduced methodology is an objective one, unlike some of the widely used cluster quality measures which may be biased in some circumstances, as argued throughout the paper. Using the validation methodology introduced by us, the comparative analysis of our results with respect to the official information presented in the considered data sets documentation outlined the performance of our approach. We have also compared our clustering algorithm with other classification techniques (even some supervised learning approaches) in terms of Accuracy, Entropy and Adjusted Rand Index, outperforming almost all of them even without outlier elimination. After eliminating the outliers, the values of some of the evaluation measures improved significantly, outperforming almost all the other approaches. We also performed comparative analysis in terms of cluster quality measures with all the other rough clustering algorithms that we know of. Even though we obtained similiar results, our algorithm is outperformed in most cases (in terms of cluster quality measures). We strongly believe, however, that this situation is caused by the fact that the considered cluster quality measures are biased in this case, favoring the other algorithms. This situation together with the lack of official information regarding hybrid data led us to propose a validation methodology which we believe is an objective one. In conclusion, we believe it is fair to say that the experiments and the comparative analysis show at least a decent performance of the proposed algorithm in comparison to other approaches. Moreover, the added value of our approach is the ability to discover hybrid data and the proposed validation methodology suggests very promising results from this perspective, as well. Future research directions will focus on improving the outlier management mechanism and a more fine-grained hybrid data handling.

CRediT author statement

Radu D. Găceanu: Methodology, conceived and designed experiments, performed experiments and provided experimental results, developed the structure and arguments of the paper, writing the manuscript and agreed with manuscript results and conclusions, reviewed and approved the final manuscript.

Arnold Szederjesi-Dragomir: Methodology, conceived and designed experiments, performed experiments and provided experimental results, writing the manuscript and agreed with manuscript results and conclusions, reviewed and approved the final manuscript.

Horia F. Pop: Methodology, writing the manuscript and agreed with manuscript results and conclusions, reviewed and approved the final manuscript.

Costel Sârbu: Methodology, writing the manuscript and agreed with manuscript results and conclusions, reviewed and approved the final manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Adam, A., & Blockeel, H. (2015). Dealing with overlapping clustering: A constraint-based approach to algorithm selection. *Proceedings of the 2015 international conference on meta-learning and algorithm selection - volume 1455* (pp. 43–54). Aachen, Germany, Germany: CEUR-WS.org.http://dl.acm.org/citation.cfm?id=3053836.3053844

Aeberhard, S., Coomans, D., & de Vel, O. (1994). Comparative analysis of statistical pattern recognition methods in high dimensional settings. *Pattern Recognition, 27*, 1065–1077.http://www.sciencedirect.com/science/article/pii/0031320394901457

Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., & Mirjalili, S. (2019). Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. *Knowledge and Information Systems*.DOI:10.1007/s10115-019-01358-x

Bagirov, A. M., Taheri, S., & Ugon, J. (2016). Nonsmooth dc programming approach to the minimum sum-of-squares clustering problems. *Pattern Recognition, 53*, 12–24. http://www.sciencedirect.com/science/article/pii/S0031320315004318

Bagirov, A. M., Ugon, J., & Webb, D. (2011). Fast modified global k-means algorithm for incremental cluster construction. *Pattern Recognition, 44*, 866–876.http://www.sciencedirect.com/science/article/pii/S0031320310005029

Bera, S., Giri, P. K., Jana, D. K., Basu, K., & Maiti, M. (2018). Multi-item 4d-tps under budget constraint using rough interval. *Applied Soft Computing, 71*, 364–385. https://doi.org/10.1016/j.asoc.2018.06.037http://www.sciencedirect.com/science/article/pii/S1568494618303715

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Norwell, MA, USA: Kluwer Academic Publishers.

Bezdek, J. C., & Pal, N. R. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 28*, 301–315.DOI:10.1109/3477.678624

Bharadwaj, A., & Ramanna, S. (2019). Categorizing relational facts from the web with fuzzy rough sets. *Knowledge and Information Systems, 61*, 1695–1713.DOI:10.1007/s10115-018-1250-6

Bryant, A., & Cios, K. (2017). Rnn-dbscan: A density-based clustering algorithm using reverse nearest neighbor density estimates. *IEEE Transactions on Knowledge and Data Engineering, 30*, 1109–1121.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P. A., Łukasik, S., & Żak, S. (2010). Complete gradient clustering algorithm for features analysis of x-ray images. In E. Pietka, & J. Kawa (Eds.), *Information technologies in biomedicine* (pp. 15–24). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, D., wu Cui, D., xue Wang, C., & rong Wang, Z. (2006). A rough set-based hierarchical clustering algorithm for categorical data. *International Journal of Information Technology, 12*, 149–159.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1*, 224–227.DOI:10.1109/TPAMI.1979.4766909

Dua, D., & Graff, C. (2017). UCI machine learning repository. http://archive.ics.uci.edu/ml.

F, G. (1989). Tabu search part i. *ORSA J Comput, 3*, 190–206.

Fisher, R. A. (1936). UCI machine learning repository: Iris data set. http://archive.ics.uci.edu/ml/datasets/Iris.

Forina, M. (1991). UCI machine learning repository: Wine data set. https://archive.ics.uci.edu/ml/datasets/wine.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning* (1st). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Gribel, D., & Vidal, T. (2019). Hg-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering. *Pattern recognition, 88*, 569–583.http://www.sciencedirect.com/science/article/pii/S0031320318304436

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations, 11*, 10–18.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Ismkhan, H. (2018). Ikmeans: An iterative clustering algorithm based on an enhanced version of the k-means. *Pattern Recognition, 79*, 402–413.http://www.sciencedirect.com/science/article/pii/S0031320318300694

Kato, Y., Saeki, T., & Mizuno, S. (2018). Considerations on the principle of rule induction by strim and its relationship to the conventional rough sets methods. *Applied Soft Computing, 73*, 933–942. https://doi.org/10.1016/j.asoc.2018.09.009http://www.sciencedirect.com/science/article/pii/S1568494618305258

Kennedy, J. (2010). *Particle swarm optimization* (pp. 760–766). Springer US.Boston, MA

Khan, K., Rehman, S. U., Aziz, K., Fong, S., & Sarasvady, S. (2014). Dbscan: Past, present and future. *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)* (pp. 232–238). IEEE.

Krishnapuram, R., & Keller, J. M. (1993). A possibilistic approach to clustering. *Trans. Fuz Sys., 1*, 98–110. https://doi.org/10.1109/91.227387

Kulczycki, P. (2012). UCI machine learning repository: Seeds data set. https://archive.ics.uci.edu/ml/datasets/seeds.

Lamirel, J., Dugue, N., & Cuxac, P. (2016). New efficient clustering quality indexes. *2016 international joint conference on neural networks (IJCNN)* (pp. 3649–3657).

Lamirel, J., Mall, R., Cuxac, P., & Safi, G. (2011). Variations to incremental growing neural gas algorithm based on label maximization. *The 2011 international joint conference on neural networks* (pp. 956–965).DOI:10.1109/IJCNN.2011.6033326

Lei, L. (2018). Wavelet neural network prediction method of stock price trend based on rough set attribute reduction. *Applied Soft Computing, 62*, 923–932. https://doi.org/10.1016/j.asoc.2017.09.029http://www.sciencedirect.com/science/article/pii/S1568494617305689

Li, Y., cong Fan, J., Pan, J. S., han Mao, G., & kun Wu, G. (2019). A novel rough fuzzy clustering algorithm with a new similarity measurement. *Journal of Internet Technology, 20*, 1145–1156.https://jit.ndhu.edu.tw/article/view/2091

Liang, B., Cai, J., & Yang, H. (2022). A new cell group clustering algorithm based on validation & correction mechanism. *Expert systems with applications, 193*, 116410. https://doi.org/10.1016/j.eswa.2021.116410https://www.sciencedirect.com/science/article/pii/S0957417421016985

Lingras, P., & West, C. (2004). Interval set clustering of web users with rough k-means. *Journal of Intelligent Information Systems, 23*, 5–16. https://doi.org/10.1023/B:JIIS.0000029668.88665.1a

Liu, Y., Qin, K., & Martinez, L. (2018). Improving decision making approaches based on fuzzy soft sets and rough soft sets. *Applied Soft Computing, 65*, 320–332. https://doi.org/10.1016/j.asoc.2018.01.012http://www.sciencedirect.com/science/article/pii/S1568494618300188

Luo, G., Li, H., Ma, B., & Wang, Y. (2022). Event-triggered adaptive fuzzy control for automated vehicle steer-by-wire system with prescribed performance: Theoretical design and experiment implementation. *Expert Systems with Applications, 195*, 116458. https://doi.org/10.1016/J.ESWA.2021.116458

Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. *In 5-th Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Maji, P., & Pal, S. K. (2007). Rough set based generalized fuzzy-means algorithm and quantitative indices. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 37*, 1529–1540. https://doi.org/10.1109/TSMCB.2007.906578

Maji, P., & Pal, S. K. (2012a). *Rough-fuzzy pattern recognition: Applications in bioinformatics and medical imaging* (1st). Wiley-IEEE Computer Society Pr.

Maji, P., & Pal, S. K. (2012b). *Rough-Fuzzy pattern recognition: Applications in bioinformatics and medical imaging* (1st). Wiley-IEEE Computer Society Pr.

Maneckshaw, B., & Mahapatra, G. S. (2022). Novel fuzzy matrix swap algorithm for fuzzy directed graph on image processing. *Expert systems with applications, 193*, 116291. https://doi.org/10.1016/J.ESWA.2021.116291

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007http://www.sciencedirect.com/science/article/pii/S0965997813001853

Onan, A. (2018a). Biomedical text categorization based on ensemble pruning and optimized topic modelling. *Computational and Mathematical Methods in Medicine, 2018*.

Onan, A. (2018b). An ensemble scheme based on language function analysis and feature engineering for text genre classification. *Journal of Information Science, 44*, 28–47.

Onan, A. (2019a). Consensus clustering-based undersampling approach to imbalanced learning. *Scientific Programming, 2019*.

Onan, A. (2019b). Topic-enriched word embeddings for sarcasm identification. *Computer science on-line conference* (pp. 293–304). Springer.

Onan, A. (2019c). Two-stage topic extraction model for bibliometric data analysis based on word embeddings and clustering. *IEEE Access, 7*, 145614–145633.

Onan, A. (2020). Mining opinions from instructor evaluation reviews: Adeep learning approach. *Computer Applications in Engineering Education, 28*, 117–138.

Onan, A. (2021a). Sentiment analysis on massive open online course evaluations: A text mining and deep learning approach. *Computer Applications in Engineering Education, 29*, 572–589.

Onan, A. (2021b). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency and Computation: Practice and Experience, 33*, e5909.

Onan, A. (2022). Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *Journal of King Saud University-Computer and Information Sciences, 34*, 2098–2117.

Onan, A., & Korukoğlu, S. (2017). A feature selection model based on genetic rank aggregation for text sentiment classification. *Journal of Information Science, 43*, 25–38.

Onan, A., Korukoğlu, S., & Bulut, H. (2016). Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications, 57*, 232–247.

Onan, A., Korukoğlu, S., & Bulut, H. (2017). A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification. *Information Processing & Management, 53*, 814–833.

Onan, A., & Toçoğlu, M. A. (2021). A term weighted neural language model and stacked bidirectional lstm based framework for sarcasm identification. *IEEE Access, 9*, 7701–7722.

Osman, I. H., & Christofides, N. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research, 1*, 317–336. https://doi.org/10.1016/0969-6016(94)90032-9http://www.sciencedirect.com/science/article/pii/0969601694900329

Pamucar, D., Stevic, Z., & Zavadskas, E. K. (2018). Integration of interval rough AHP and interval rough MABAC methods for evaluating university web pages. *Applied Soft Computing, 67*, 141–163. https://doi.org/10.1016/j.asoc.2018.02.057http://www.sciencedirect.com/science/article/pii/S1568494618301194

Patel, K., Vala, J., Pandya, J., & Nagar, V. V. (2014). Comparison of various classification algorithms on iris datasets using weka.

Pawlak, Z. (1992). *Rough sets: Theoretical aspects of reasoning about data*. Norwell, MA, USA: Kluwer Academic Publishers.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research : JMLR, 12*, 2825–2830.http://dl.acm.org/citation.cfm?id=1953048.2078195

Pop, H. F., & Sârbu, C. (1996). A new fuzzy regression algorithm. *Analytical Chemistry, 68*, 771–778. https://doi.org/10.1021/ac950549uPMID: 21619171

Pop, H. F., Sârbu, C., Horowitz, O., & Dumitrescu, D. (1996). A fuzzy classification of the chemical elements. *Journal of Chemical Information and Computer Sciences, 36*, 465–482. https://doi.org/10.1021/ci9502717

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, 66*, 846–850.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science (New York, N.Y.), 344*, 1492–1496.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7http://www.sciencedirect.com/science/article/pii/0377042787901257

S, F. R., & P, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2*, 559–572.DOI:10.1080/14786440109462720

Sakhardande, M. J., & Gaonkar, R. S. P. (2022). On solving large data matrix problems in fuzzy AHP. *Expert Systems with Applications, 194*, 116488. https://doi.org/10.1016/J.ESWA.2021.116488

Sârbu, C., Horowitz, O., & Pop, H. F. (1996). A fuzzy cross-classification of the chemical elements, based on their physical, chemical, and structural features. *Journal of Chemical Information and Computer Sciences, 36*, 1098–1108. https://doi.org/10.1021/ci960050g

School, E. (2012). Benchee. https://elixirschool.com.https://elixirschool.com/en/lessons/libraries/benchee/.

Scott, S. M., O'Hare, W. T., & Ali, Z. (2003). *Fuzzy logic and fuzzy classification techniques* (pp. 95–134). Springer Berlin Heidelberg.Berlin, Heidelberg

Shang, Z., Yang, X., Barnes, D., & Wu, C. (2022). Supplier selection in sustainable supply chains: using the integrated bwm, fuzzy shannon entropy, and fuzzy multimoora methods. *Expert Systems with Applications, 195*, 116567. https://doi.org/10.1016/J.ESWA.2022.116567https://linkinghub.elsevier.com/retrieve/pii/S0957417422000653

Shannon, C. E. (2001). A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev., 5*, 3–55. https://doi.org/10.1145/584091.584093

Siegel, S., & Castellan, N. (1988). Nonparametric statistics for the behavioural sciences, 2nd edn mcgraw-hill book company. *New York*, 144–151.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–359. https://doi.org/10.1023/A:1008202821328

Szederjesi-Dragomir, A., Gaceanu, R. D., Pop, H. F., & Sarbu, C. (2020). Experiments on rough sets clustering with various similarity measures. *IPSI BGD TRANSACTIONS ON INTERNET RESEARCH, 16*, 75–83.

Szederjesi-Dragomir, A., Găceanu, R. D., Pop, H. F., & Sârbu, C. (2019). A comparison study of similarity measures in rough sets clustering. *2019 IEEE 15th international scientific conference on informatics* (pp. 000037–000042).DOI:10.1109/Informatics47936.2019.9119264

Viswanath, P., & Pinkesh, R. (2006). L-dbscan : A fast hybrid density based clustering method. *18th international conference on pattern recognition (ICPR'06)* (pp. 912–915). DOI:10.1109/ICPR.2006.741

Wang, P. C., Su, C. T., Chen, K. H., & Chen, N. H. (2011). The application of rough set and mahalanobis distance to enhance the quality of OSA diagnosis. *Expert systems with applications, 38*, 7828–7836.http://www.sciencedirect.com/science/article/pii/S0957417410014855

Wooldridge, M. (2009). *An introduction to multiagent systems* (2nd). Wiley Publishing.

Xie, X., Qin, X., Yu, C., & Xu, X. (2018). Test-cost-sensitive rough set based approach for minimum weight vertex cover problem. *Applied Soft Computing, 64*, 423–435. https://doi.org/10.1016/j.asoc.2017.12.023http://www.sciencedirect.com/science/article/pii/S1568494617307482

Yang, H. H., & Wu, C. L. (2009). Rough sets to help medical diagnosis - evidence from a Taiwan's clinic. *Expert Systems with Applications, 36*, 9293–9298.http://www.sciencedirect.com/science/article/pii/S0957417408008786

Yang, J., Yecies, B., Ma, J., & Li, W. (2022). Sparse fuzzy classification for profiling online users and relevant user-generated content. *Expert Systems with Applications, 194*, 116497. https://doi.org/10.1016/J.ESWA.2021.116497

Zadeh, L. (1965). Fuzzy sets. *Information and Control, 8*, 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Radu Dan Găceanu is a lecturer at the Department of Computer Science, Faculty of Mathematics and Computer Science from the Babes-Bolyai University of Cluj-Napoca, Romania. He has received the Ph.D. degree in Computer Science in 2012 and his main research interests are pattern recognition and soft computing.

Arnold Szederjesi-Dragomir is a Ph.D. student at the Department of Computer Science, Faculty of Mathematics and Computer Science from the Babes-Bolyai University of Cluj-Napoca, Romania. His main research interests are pattern recognition and agent-based computing.

Horia Florin Pop is a professor at the Department of Computer Science, Faculty of Mathematics and Computer Science from the Babes-Bolyai University of Cluj-Napoca, Romania. He published more than 125 papers in prestigious journals and conference proceedings. His research interests include soft computing and pattern recognition.

Costel Sârbu is a professor at the Department of Computer Science, Faculty of Mathematics and Computer Science from the Babes-Bolyai University of Cluj-Napoca, Romania. As per Google Scholar, he has over 2600 citations and h-index of 26. His research interests include chemometrics and soft computing.