

## Rough cognitive ensembles

Peer-reviewed author version

NAPOLES RUIZ, Gonzalo; Falcon, Rafael; PAPAGEORGIOU, Elpiniki; Bello, Rafael & VANHOOF, Koen (2017) Rough cognitive ensembles. In: INTERNATIONAL JOURNAL OF APPROXIMATE REASONING, 85, p. 79-96.

DOI: 10.1016/j.ijar.2017.03.011

Handle: <http://hdl.handle.net/1942/23693>

# Rough Cognitive Ensembles

Gonzalo Nápoles<sup>a,b</sup>, Rafael Falcon<sup>c</sup>, Elpiniki Papageorgiou<sup>d,a</sup>,  
Rafael Bello<sup>b</sup>, Koen Vanhoof<sup>a</sup>

<sup>a</sup>*Faculty of Business Economics, Hasselt Universiteit, Belgium*

<sup>b</sup>*Department of Computer Science, Universidad Central de Las Villas, Cuba*

<sup>c</sup>*School of Electrical Engineering and Computer Science, University of Ottawa, Canada*

<sup>d</sup>*Department of Computer Engineering, Technological Education Institute of Central Greece, Greece*

---

## Abstract

*Rough Cognitive Networks* are granular classifiers stemming from the hybridization of Fuzzy Cognitive Maps and Rough Set Theory. Such cognitive neural networks attempt to quantify the impact of rough granular constructs (i.e., the positive, negative and boundary regions of a target concept) over each decision class for the problem at hand. In rough classifiers, determining the precise granularity level is crucial to compute high prediction rates. Regrettably, learning the similarity threshold parameter requires reconstructing the information granules, which may be time-consuming. In this paper, we put forth a new multiclassifier system classifier named *Rough Cognitive Ensembles*. The proposed ensemble employs a collection of Rough Cognitive Networks as base classifiers, each operating at a different granularity level. This allows suppressing the requirement of learning a similarity threshold. We evaluate the granular ensemble with 140 traditional classification datasets using different heterogeneous distance functions. After comparing the proposed model to 15 well-known classifiers, the experimental evidence confirms that our scheme yields very promising classification rates.

*Key words:* machine learning, granular computing, rough set theory, fuzzy cognitive maps, rough cognitive networks, ensemble learning

---

*Email addresses:* [gonzalo.napoles@uhasselt.be](mailto:gonzalo.napoles@uhasselt.be) (Gonzalo Nápoles)

## 1. Introduction

*Pattern classification* [1] is one of the most ubiquitous real-world problems and certainly one at which humans really excel. It consists of identifying the right category (among those in a predefined set) to which an observed pattern belongs. These patterns are often described by a set of predictive attributes of numerical and/or nominal nature. Formally speaking, the pattern classification problem [1] is about building a mapping  $f : \mathcal{U} \rightarrow \mathcal{D}$  that assigns to each instance  $x \in \mathcal{U}$  described by the attribute set  $\Psi = \{\psi_1 \dots, \psi_M\}$  a decision class  $D_k$  from the  $K$  possible ones in  $\mathcal{D} = \{D_1, \dots, D_K\}$ . The mapping is learned in a *supervised* fashion, i.e., by relying on an existing set of previously labeled examples that is used to train the classification model. The learning process is regularly driven by the minimization of a cost/error metric. More complex classification problems include learning in presence of class imbalance [2] [3], class noise [4], partially labeled data [5] or multiple decision classes per object [6] [7] [8], among other scenarios.

The literature on classification models (henceforth simply called “*classifiers*”) is vast and offers a myriad of techniques that approach the classification problem from multiple angles. Decision trees [9], neural networks [10], rule-based models [11], Bayesian networks [12], support vector machines [13] or  $k$ -nearest neighbors learners [14] stand among the most popular classifiers, each having its own inherent advantages and limitations.

A well-established trend within Machine Learning (ML) is that of combining the output of multiple base classifiers to predict the class label of a given instance. These models are called *multiclassifier systems* [15] and often perform better than any single classifier, especially if some diversity (either parametric or structural) is present among the set of base classifiers. This heterogeneity may come in the form of independent samples of the training data (bootstrapping), multiple parametric configurations, different types of base classifiers used as building blocks, etc. *Bayesian voting*, *bagging*, *boosting* and *stacking* are common ensemble creation methods [16]. AdaBoost [17] and random forests [18] are well-known ensemble learners that have shown great promise in solving pattern classification problems.

A recent survey [16] explores trends in this field, including: (a) using multiobjective optimization algorithms to derive several structural elements of the ensembles; (b) employing decomposition techniques; (c) resorting to negative correlation learning schemes; (d) incorporating elements from fuzzy logic and multiple-kernel learning and (e) endowing the models with deep

38 learning features. Other reviews that focus on either a set of techniques or a  
39 particular domain can be found in the literature as well. For example, Zhao  
40 et al. [19] revised existing neural network ensembles whereas Yang et. al.  
41 [20] related ensemble approaches used in bionformatics.

42 Another ongoing ML avenue has led to the development of *granular clas-*  
43 *sifiers* [21] [22] [23] [24] [25] [26]. These are classification schemes that abide  
44 by the principles of *Granular Computing* [27] [28] [29], an emerging paradigm  
45 that relies on more symbolic constructs such as sets, intervals or similarity  
46 classes to replace numerical data as the underlying knowledge representation  
47 and reasoning vehicle. Granular classifiers offer important advantages over  
48 traditional ML techniques such as (1) a more human-centric manner of in-  
49 teracting with real-world information and its inherent uncertainty and (2)  
50 the ability to deal with the Big Data phenomenon by processing the deluge  
51 of raw data at higher levels of abstraction.

52 Recently, Nápoles and collaborators [30] introduced the notion of *rough*  
53 *cognitive mapping* in the pattern recognition context. Rough Cognitive Net-  
54 works (RCNs) are granular classifiers featuring a synergy between Fuzzy  
55 Cognitive Maps (FCMs) [31] and Rough Set Theory (RST) [32]. Being more  
56 explicit, an RCN can be defined as a sigmoid FCM whose input concepts  
57 denote information granules (namely, the RST-derived positive, boundary  
58 and negative regions of the decision classes) whereas output concepts rep-  
59 resent decision classes of the problem at hand. This granular model seems  
60 especially suitable to handle inconsistent data given the approximation space  
61 constructed over each output concept (decision class) by means of the three  
62 RST-based regions induced by a similarity relation. It is worth mentioning  
63 that two objects are considered inconsistent if their similarity degree exceeds  
64 a certain threshold, but they lead to different decisions.

65 RCN's classification performance has been found competitive with respect  
66 to state-of-the-art classifiers [33] [34] [30]. Despite these encouraging results,  
67 RCN-based models are still very sensitive to an input parameter denoting  
68 the similarity threshold upon which the rough information granules are built.  
69 Determining the exact granularity degree is not trivial, since higher values do  
70 not necessarily lead to optimal prediction rates. To overcome this drawback,  
71 Nápoles et. al. [30] proposed a hyperparameter learning method to estimate  
72 the similarity threshold value from historical data. Nevertheless, evaluating a  
73 candidate solution implies reconstructing the granular regions from scratch,  
74 which is time-consuming in large datasets.

75 In this paper, we build upon the RCNs formalism by developing a *gran-*

76 *ular ensemble* model in order to deal with the RCN’s parametric sensitivity.  
77 Rough Cognitive Ensembles (RCEs) use a collection of RCNs as base classi-  
78 fiers, each operating at a different granularity level. It is worth mentioning  
79 that the concept of *granular ensembles* seems to still be in its infancy though.  
80 A good starting point is the use of granular classifiers as base classifiers, given  
81 the momentum they are presently enjoying. The model proposed in this pa-  
82 per follows this idea, and it may well be one of the first granular ensemble  
83 techniques ever put forth. As a result, the hyperparameter learning method  
84 is no longer required, which notably increases the practical usability of this  
85 novel granular classifier in real-world applications.

86 The second contribution of this paper is oriented to the exhaustive evalu-  
87 ation of rough cognitive classifiers in presence of 140 ML datasets. As far as  
88 we know, RCNs have been applied to a few real-life problems, but there is no  
89 record of their performance in solving standard ML problems. This evalua-  
90 tion includes studying the algorithm’s performance for several heterogeneous  
91 distance functions. After comparing the ensemble model to 15 state-of-the-  
92 art ML algorithms, the experimental evidence suggests that the proposed  
93 classifier produces highly competitive prediction rates.

94 The rest of this paper is structured as follows. Section 2 goes over im-  
95 portant concepts that are related to this study such as *rough sets*, *three-way*  
96 *decision rules* and *fuzzy cognitive maps*. Section 3 emphasizes on the building  
97 blocks of the granular ensemble, Section 4 presents the foundations of the  
98 proposed ensemble classifier, while Section 5 describes three-widely used het-  
99 erogeneous distance functions. Moreover, Section 6 introduces the empirical  
100 simulations and its ensuing discussion. Section 7 formalizes the concluding  
101 remarks and future research directions to be explored.

## 102 2. Theoretical background

103 This Section sheds light on several important concepts to this study,  
104 namely classical and extended rough sets, foundations of three-way decision  
105 rules, cognitive mapping and granular cognitive maps.

### 106 2.1. Rough Set Theory

107 RST is a methodology proposed in the early 1980’s for handling uncer-  
108 tainty that is manifested in the form of inconsistent data [32] [35] [36]. Let  
109  $DS = (\mathcal{U}, \Psi \cup \{d\})$  denote a *decision system* where  $\mathcal{U}$  is a non-empty fi-  
110 nite set of objects called the universe of discourse,  $\Psi$  is a non-empty finite

111 set of attributes describing any object in  $U$  and  $d \notin \Psi$  represents the deci-  
 112 sion attribute. Any subset  $X \in \mathcal{U}$  can be approximated by two crisp sets,  
 113 which are referred to as its *lower* and *upper approximations* and denoted by  
 114  $\underline{\Phi}X = \{x \in \mathcal{U} \mid [x]_{\Phi} \in X\}$  and  $\overline{\Phi}X = \{x \in \mathcal{U} \mid [x]_{\Phi} \cap X \neq \emptyset\}$ , respectively.  
 115 In this classic formulation, the equivalence class  $[x]_{\Phi}$  comprises the set of ob-  
 116 jects in  $\mathcal{U}$  that are deemed inseparable from  $x$  according to the information  
 117 contained in the attribute subset  $\Phi \subseteq \Psi$ .

118 The lower and upper approximations are the basis for computing the  
 119 positive, negative and boundary regions of any set  $X$ . The *positive region*  
 120  $POS(X) = \underline{\Phi}X$  includes those objects that are certainly contained in  $X$ ; the  
 121 *negative region*  $NEG(X) = \mathcal{U} - \overline{\Phi}X$  denotes those objects that are certainly  
 122 not related to  $X$ , while the *boundary region*  $BND(X) = \overline{\Phi}X - \underline{\Phi}X$  captures  
 123 the objects whose membership to the set  $X$  is uncertain, i.e., they might  
 124 be members of  $X$ . These three rough regions comprise *information granules*  
 125 upon which the RCN leans to map its set of input-type neurons and produce  
 126 conclusions, as explained later in Section 3.

### 127 2.1.1. Using similarity relations in RST

128 In the classical RST formulation, two objects are considered inseparable  
 129 w.r.t.  $\Phi \subseteq \Psi$  if they have identical values for all attributes. This definition is  
 130 adequate for nominal attributes but is too rigid when dealing with numerical  
 131 ones, given that marginal differences between two numerical values could toss  
 132 two nearly identical objects into different inseparability classes. To counter  
 133 this stringent definition, the equivalence requirement on the inseparability  
 134 relation  $R$  is relaxed by resorting to *similarity relations*.

135 Equation (1) shows an indiscernibility relation, where  $0 \leq S(x, y) \leq 1$  is  
 136 a similarity function. This weaker binary relation states that objects  $x$  and  
 137  $y$  are deemed inseparable as long as their similarity degree  $S(x, y)$  exceeds a  
 138 similarity threshold  $0 \leq \xi \leq 1$ . This user-specified parameter determines the  
 139 granularity degree; therefore, its precise estimation may become essential for  
 140 designing high-performing rough classifiers.

$$R : xRy \iff S(x, y) \geq \xi \tag{1}$$

141 The similarity function could be formulated in a variety of ways. In this  
 142 study, we assume that  $S(x, y) = 1 - \delta(x, y)$ , where  $\delta(x, y)$  denotes the distance  
 143 between objects  $x$  and  $y$ . As a further contribution, this paper investigates  
 144 the effect of using different distance functions over the overall prediction rates  
 145 computed by the proposed rough cognitive classifiers.

146 It should be noticed that the similarity relation  $R$  does not induce a  
 147 *partition* of  $\mathcal{U}$  into a set of equivalence classes but rather a *covering* of  $\mathcal{U}$   
 148 into multiple similarity classes  $\bar{R}(x)$ . This suggests that an object  $x$  could  
 149 simultaneously belong to more than one similarity class at the same time.  
 150 Next, we revise the theoretical foundations of three-way decision rules, which  
 151 comprise the core of rough cognitive mapping.

### 152 2.1.2. The three-way-decisions model

153 Yao [37] introduced the *three-way decision rules* based on three disjoint  
 154 regions produced by the lower and upper approximations, namely, the posi-  
 155 tive region  $POS(X)$ , the boundary region  $BND(X)$ , and the negative region  
 156  $NEG(X)$ . By exploiting the positive and negative regions we can extract  
 157 confident decisions while including the boundary regions in the reasoning  
 158 process allows concluding an alternative decision that is neither positive nor  
 159 negative. These classification rules are depicted as follows:

- 160 •  $Des([x]_{\Phi}) \rightarrow_P Des(X)$ , for  $[x]_{\Phi} \subseteq POS(X)$
- 161 •  $Des([x]_{\Phi}) \rightarrow_B Des(X)$ , for  $[x]_{\Phi} \subseteq BND(X)$
- 162 •  $Des([x]_{\Phi}) \rightarrow_N Des(X)$ , for  $[x]_{\Phi} \subseteq NEG(X)$

163 In these rules  $Des([x]_{\Phi})$  represents the logic formula defining the equiv-  
 164 alence class  $[x]_{\Phi}$  and  $Des(X)$  is the name of the concept. A *positive rule*  
 165 is used for accepting, a *negative rule* for rejecting and a *boundary rule* for  
 166 abstaining. More precisely, if  $[x]_{\Phi} \subseteq POS(X)$ , we accept  $x$  to be a mem-  
 167 ber of the concept  $X$ . If  $[x]_{\Phi} \subseteq NEG(X)$ , we reject  $x$  to be a member of  
 168  $X$ . If  $[x]_{\Phi} \subseteq BND(X)$ , we neither accept nor reject  $x$  to be a member of  
 169  $X$ ; instead, we make a decision of deferment, abstaining or non-commitment  
 170 [38]. Such rules model our inability to make a definite acceptance or rejection  
 171 decision in scenarios with insufficient information.

172 Within the classic rough set model, the above classification rules are  
 173 unnecessarily restrictive, but one can generalize these rules to avoid such  
 174 limitations. The studies in [39] and [38] introduced a general probabilistic  
 175 approach, called *Decision-theoretic Rough Set model*, which uses two states  
 176 and three actions to characterize the decision process.

177 As mentioned before, the abstract semantic of three-way decision rules is  
 178 used for automatically constructing the RCN topology. This allows hybridiz-  
 179 ing RST elements with the neural reasoning mechanism of fuzzy cognitive  
 180 mapping in an attempt to design an effective granular classifier.

181 *2.2. Fuzzy Cognitive Maps*

182 Without lack of generality, FCMs can be defined as interpretable recurrent  
 183 neural networks widely used in modeling and simulation purposes [31]. They  
 184 consist of a set of concepts (i.e., objects, variables or entities for a particular  
 185 problem) and their causal relations. The activation value of such concepts  
 186 (also called neurons) regularly takes values in the  $[0, 1]$  interval. The higher  
 187 the activation value of a neuron, the stronger its influence over the system  
 188 under investigation. On the other hand, the strength of the causal relation  
 189 between two concepts  $C_i$  and  $C_j$  is quantified by a weight  $w_{ij} \in [-1, 1]$  and  
 190 denoted via a directed edge from  $C_i$  to  $C_j$ . There are three possible types of  
 191 causal relationships among neural processing entities that express the type  
 192 of influence from one neuron to the others:

- 193 • If  $w_{ij} > 0$  then an increment (decrement) in the cause  $C_i$  produces an  
 194 increment (decrement) of the effect  $C_j$  with intensity  $|w_{ij}|$ .
- 195 • If  $w_{ij} < 0$  then an increment (decrement) in the cause  $C_i$  produces a  
 196 decrement (increment) of the neuron  $C_j$  with intensity  $|w_{ij}|$ .
- 197 • If  $w_{ij} = 0$  then there is no causal relation between  $C_i$  and  $C_j$ .

198 The activation update rule in an FCM is displayed in Equation (2), with  
 199  $A^{(0)}$  being the initial activation vector. This rule is iteratively repeated until  
 200 a stop condition is met. A new activation vector is calculated at each discrete  
 201 time step  $t$ . After a fixed number of iterations, the map will arrive at one of  
 202 the following states: (i) equilibrium point, (ii) limited cycle or (iii) chaotic  
 203 behavior [40]. The map is said to have converged if it reaches a fixed-point  
 204 attractor. Otherwise, the process terminates after a maximum number of  
 205 iterations  $T$  is reached. Whichever the case, the FCM output corresponds to  
 206 the response vector at the last discrete time step.

$$A_i^{(t+1)} = f \left( \sum_{j=1}^M w_{ji} A_j^{(t)} + A_i^{(t)} \right), i \neq j \quad (2)$$

207 The function  $f(\cdot)$  in Equation (2) denotes a monotonically non-decreasing  
 208 nonlinear function used to clamp the activation value of each neuron to the  
 209 allowed interval. Examples of such functions are the bivalent function, the  
 210 trivalent function and the sigmoid variants [41]. In this study, we confine  
 211 ourselves to using a sigmoid function  $f(A_i) = 1/(1 + \exp(-\lambda A_i))$  since it has



212 exhibited superior prediction capabilities [41]. In the sigmoid function, the  
213 parameter  $\lambda > 0$  denotes the steepness factor and influences the sensitivity of  
214 the neuron. Nápoles et al. [30] empirically concluded that, in the framework  
215 of rough cognitive mapping,  $2 \leq \lambda \leq 5$  are suitable values.

216 FCMs have been augmented with different types of information granules,  
217 thus giving rise to high-level constructs known as *Granular Cognitive Maps*  
218 (GCMs). Pedrycz [42] mentions the allocation of information granularity as  
219 a pivotal driving force behind the development of these types of granular  
220 structures and describes five protocols as its realization mechanisms. For  
221 example, Pedrycz [43] and his collaborators [44] put forth a granular repre-  
222 sentation of time series in which FCM nodes are generated after the cluster  
223 prototypes induced by fuzzy  $c$ -means [45] over the space of amplitude and  
224 change of amplitude. More recently, Nápoles et. al. [8] extended this model  
225 to solve graded multilabel classification problems.

226 Homenda et. al. [46] adopted numeric intervals as the granulation vehicle  
227 for their GCM weight matrix. As a result, each FCM weight is no longer a  
228 number but an interval. The authors elaborate on three methodologies for  
229 building a GCM from scratch by maintaining an adequate balance between  
230 specificity and generality in the design of the interval-based FCM weights  
231 and the ensuing map operations. They found the resulting GCM had a good  
232 degree of coverage without a loss in precision.

233 Another development in the realm of GCMs is that of RCNs, which is  
234 the underlying building block for the proposed model. Section 3 elaborates  
235 on RCNs and stresses their advantages and limitations.

### 236 3. Rough Cognitive Networks

237 Nápoles et. al. [33] [34] [30] recently introduced the concept of *rough cog-*  
238 *nitive mapping*, which hybridizes RST and FCMs. In this granular model,  
239 the abstract semantics of three-way decision rules is used to define the FCM  
240 topology. This allows overcoming the limitations in the expression and archi-  
241 tecture of traditional cognitive mapping as the network topology (i.e., map  
242 concepts and causal relations) are directly computed from historical data,  
243 removing the the expert intervention requirements. Definition 1 formalizes  
244 the semantics behind this granular cognitive classifier.

245 **Definition 1.** *A Rough Cognitive Network  $\mathcal{R}_{(\xi_i)}$  is a sigmoid FCM whose*  
246 *input neurons represent rough approximation regions and the output neurons*  
247 *denote the set of class labels for a pattern classification problem.*

248 The first step when constructing an RCN is related to the *granulation*  
 249 *of the input data* using RST. The positive, boundary and negative regions  
 250 of each decision class according to a predefined attribute subset  $\Phi \subseteq \Psi$  are  
 251 computed using the training dataset and a predefined similarity relation  $R$   
 252 to define the similarity classes (see Section 2.1). The similarity relations may  
 253 employ either the whole attribute set  $\Psi$  or a subset  $\Phi \subseteq \Psi$ , in case a feature  
 254 selection technique has identified one as such.

255 It is important to notice that when solving ML problems with numerical  
 256 attributes, the similarity class of a new instance  $\bar{R}(x)$  might activate  
 257 multiple decision rules. If  $\bar{R}(x)$  only comprises instances related to a single  
 258 decision class, then the similarity class will be completely contained within  
 259 a specific decision region. If the similarity class activates the positive region  
 260 of multiple decisions, the RCN will calculate confidence levels for each active  
 261 rule and then compute the preference degree associated with each decision.  
 262 In more complex scenarios where only negative and boundary regions are  
 263 activated, RCNs are capable of using the knowledge about positive, negative  
 264 and boundary regions for computing the decision class.

265 The second step when constructing an RCN is the *automated topology*  
 266 *design*, where a sigmoid FCM is automatically created from the aforementioned  
 267 RST-based information granules. In this scheme, each rough region  
 268 is mapped to an input-type neural entity whereas each decision class gives  
 269 rise to an output-type concept. Thus, the RCN graph will comprise at most  
 270  $3|\mathcal{D}| + |\mathcal{D}| = 4|\mathcal{D}|$  neurons, with  $\mathcal{D} = \{D_1, \dots, D_K\}$  being the set of decision  
 271 classes. It should be stated that output neurons do not influence other  
 272 neurons since they are receiver concepts and are only used to compute the  
 273 response vector. Rules  $R_1 - R_4$  formalize the direction and intensity of the  
 274 causal weights in the proposed topology; these weights are estimated by using  
 275 the abstract semantics of three-way decision rules.

- 276 •  $(R_1)$  IF  $C_i$  is  $P_k$  AND  $C_j$  is  $D_k$  THEN  $w_{ij} = 1.0$
- 277 •  $(R_2)$  IF  $C_i$  is  $P_k$  AND  $C_j$  is  $D_{v \neq k}$  THEN  $w_{ij} = -1.0$
- 278 •  $(R_3)$  IF  $C_i$  is  $P_k$  AND  $C_j$  is  $P_{v \neq k}$  THEN  $w_{ij} = -1.0$
- 279 •  $(R_4)$  IF  $C_i$  is  $N_k$  AND  $C_j$  is  $D_k$  THEN  $w_{ij} = -1.0$

280 In such rules,  $C_i$  and  $C_j$  represent two map neurons,  $P_k$  and  $N_k$  are the  
 281 positive and negative regions related to the  $k$ th decision respectively while

282  $-1 \leq w_{ij} \leq 1$  is the causal weight between the cause  $C_i$  and the effect  $C_j$ .  
 283 The reader may observe that neurons can be grouped into four categories  
 284 (positive, negative, boundary and decision neurons) which allows designing  
 285 the network using the principles behind three-way decision rules.

286 Although the boundary regions are concerned with an abstaining decision,  
 287 an instance  $x \in BND(X_k)$  could be positively related to the  $k$ th decision  
 288 alternative. Therefore, an additional rule considering the knowledge about  
 289 boundary regions is introduced:

- 290 • ( $R_5$ ) IF  $C_i$  is  $B_k$  AND  $C_j$  is  $D_v$  AND  $BND(X_k) \cap BND(X_v) \neq \emptyset$   
 291 THEN  $w_{ij} = 0.5$

292 The last step is related to the *network exploitation* on which the response  
 293 vector  $\mathcal{A}_x(\mathcal{D}) = [A_x(D_1), \dots, A_x(D_k), \dots, A_x(D_K)]$  is computed. The input  
 294 pattern  $x$  is presented to the granular classifier as an input vector  $A^{(0)}$  that  
 295 allows activating the causal network. Rules  $R_6 - R_8$  formalize the method  
 296 used to activate the input neurons, which is based on the inclusion degree of  
 297 the pattern to each rough granular region.

298 • ( $R_6$ ) IF  $C_i$  is  $P_k$  THEN  $A_i^{(0)} = \frac{|\bar{R}(x) \cap POS(X_k)|}{|POS(X_k)|}$

299 • ( $R_7$ ) IF  $C_i$  is  $N_k$  THEN  $A_i^{(0)} = \frac{|\bar{R}(x) \cap NEG(X_k)|}{|NEG(X_k)|}$

300 • ( $R_8$ ) IF  $C_i$  is  $B_k$  THEN  $A_i^{(0)} = \frac{|\bar{R}(x) \cap BND(X_k)|}{|BND(X_k)|}$

301 Once the excitation vector  $A^{(0)}$  has been computed, the reasoning rule  
 302 depicted in Equation (2) is performed until either the network converges to  
 303 a fixed-point or a maximal number of iterations is reached. Next, the class  
 304 with the highest activation value is assigned to the pattern.

305 Figure 1 displays an RCN for solving two-class pattern classification prob-  
 306 lems. Notice that we added a self-reinforcement positive causal connection  
 307 to each input neuron with the goal of preserving its initial excitation level  
 308 when performing the neural updating rule.

309 Rough cognitive mapping brings up the following advantages:

- 310 1. It allows handling pattern classification problems with both numerical  
 311 and nominal attributes, since the rough regions are constructed on the  
 312 basis of weak binary (similarity) relations.

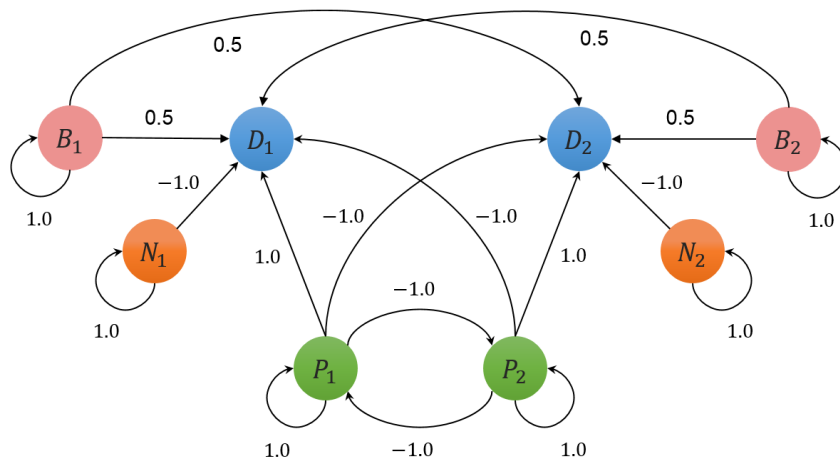


Figure 1: RCN for pattern recognition problems with two decision classes.

- 313 2. It yields a numeric vector of preference degrees associated to decision  
 314 classes instead of binary decisions for each alternative, which is often a  
 315 desirable feature in decision-making scenarios.
- 316 3. The RCN topology is learned from historical data, thus freeing the user  
 317 from an entangled and likely subjective modeling phase to derive the  
 318 causal relationships among the different concepts.
- 319 4. The granulation step allows reasoning at a higher level of abstraction,  
 320 thus reducing the dimensionality of the feature space. Consequently, the  
 321 number of neural processing entities in the causal network does not scale  
 322 up with the number of features describing the objects, which facilitates  
 323 the processing of Big Dimensionality problems.

#### 324 4. Rough Cognitive Ensembles

325 A key issue when constructing an RCN is related to the proper selection of  
 326 the similarity threshold. This parameter defines whether two objects are sim-  
 327 ilar or not, which then influences the creation of the similarity classes upon  
 328 which the rough approximations (RCN input nodes) are built. Regrettably,  
 329 the RCN performance is highly sensitive to this user-specified parameter,  
 330 thus small variations on the granularity degree may lead to quite different  
 331 outcomes. As an alternative, Nápoles et. al. [30] proposed a hyperparame-  
 332 ter learning method to estimate the similarity threshold value from historical  
 333 data. The main drawback of this procedure relies on its computational com-

334 plexity due to the fact that a single objective function evaluation requires re-  
 335 calculating the lower and upper approximations of all decision classes, which  
 336 could be time-consuming for large-scale datasets.

337 Let us assume that  $\mathcal{U}_1 \subset \mathcal{U}$  is the training set and  $\mathcal{U}_2 \subset \mathcal{U}$  is the held-out  
 338 test (validation) set such that  $\mathcal{U}_1 \cap \mathcal{U}_2 = \emptyset$ . The computational complexity  
 339 of building the upper and lower approximations is  $O(|\Phi||\mathcal{U}_1|^2)$ , with  $\Phi$  being  
 340 the attribute set, whereas the complexity of building the network topology  
 341 is  $O(|\mathcal{D}|^2)$ , with  $\mathcal{D}$  being the set of decision classes. Besides, the complexity  
 342 of exploiting the granular network for  $|\mathcal{U}_2|$  instances is  $O(|\mathcal{U}_2||\Phi||\mathcal{U}_1|^2)$ . This  
 343 implies that the temporal complexity of evaluating a single parameter value  
 344 is  $O(\max\{|\Phi||\mathcal{U}_1|^2, |\mathcal{D}|^2, |\mathcal{U}_2||\Phi||\mathcal{U}_1|^2\})$ . Due to the fact that  $|\mathcal{U}_1| \geq |\mathcal{U}_2|$  in  
 345 most machine learning scenarios, we can conclude that the overall complexity  
 346 of this learning method is  $O(T|\Phi||\mathcal{U}_1|^3)$ , where  $T$  is the number of cycles.  
 347 Unfortunately, this may negatively affect the practical usability of RCNs in  
 348 solving real-world pattern classification problems.

349 On the other hand, estimating a suitable similarity threshold does not  
 350 necessarily ensure that the granular network will produce optimal predictions.  
 351 This happens because the validation and test sets must be disjoint, and so the  
 352 parameter learning method fits the model to the examples in the validation  
 353 set. Moreover, it might occur that  $\bar{R}(x) = \emptyset$  for some testing object as a  
 354 result of estimating a very strict threshold for validation examples. If so, the  
 355 model uses the  $k$ -nearest neighbors to activate the network.

356 Next, we introduce the notion of Rough Cognitive Ensembles (RCEs) in  
 357 an attempt to get rid of the parameter learning scheme while increasing the  
 358 performance of rough cognitive mapping.

#### 359 4.1. The ensemble architecture

360 Informally speaking, an RCE is an ensemble of several RCNs, where each  
 361 individual granular network operates at a different granularity degree. To in-  
 362 duce different granularity degrees, we use a random similarity threshold when  
 363 constructing the three approximation regions associated with each base clas-  
 364 sifier. This approach attempts to eliminate the parameter learning require-  
 365 ment, which in fact does not necessarily lead to optimal prediction rates when  
 366 testing the classifier. Definition 2 summarizes the semantics of the granular  
 367 cognitive ensembles presented in this research.

368 **Definition 2.** A Rough Cognitive Ensemble  $\Gamma_{(\mathcal{R})}$  is a multiclassifier system  
 369 composed of a set  $\mathcal{R} = \{\mathcal{R}_{(\xi_1)}, \dots, \mathcal{R}_{(\xi_i)}, \dots, \mathcal{R}_{(\xi_N)}\}$  of  $N$  different RCNs as  
 370 base classifiers, where the  $i$ th granular cognitive network  $\mathcal{R}_{(\xi_i)}$  is constructed  
 371 from a (randomly) selected granularity degree.

372 Another important aspect when designing a classifier ensemble is related  
 373 to the aggregation of multiple outputs. Combining the decisions of different  
 374 models means amalgamating the various outputs into a single prediction.  
 375 The simplest way to do this in classification models is adopting a standard  
 376 (or weighted) voting scheme. Voting gives a significant improvement in clas-  
 377 sification accuracy and stability [48]. The higher the number of base classi-  
 378 fiers, the merrier: predictions made by voting become more reliable as more  
 379 votes are taken into account. Therefore, the classification result for each test  
 380 instance is obtained by voting on all  $N$  classifiers.

381 Theoretical and empirical results [49] [50][18] have shown that combining  
 382 multiple base classifiers leads to optimal performance if these classifiers are  
 383 not strongly correlated with one another. In the proposed granular ensemble,  
 384 the diversity among base classifiers is promoted by using different similarity  
 385 thresholds  $[\xi_1, \dots, \xi_i, \dots, \xi_N]$  instead of using a single value.

386 Figure 2 displays an RCE comprised of  $N$  base classifiers with  $K$  decision  
 387 classes, where  $D_k^{(i)}$  denotes the  $k$ th decision class for the  $i$ th granular network  
 388  $\mathcal{R}_{(\xi_i)}$  and  $D_k$  is the aggregated-type concept associated with the  $k$ th decision  
 389 class. In this architecture, the function  $A(\cdot)$  is used to compute the activation  
 390 value of all sigmoid neurons involved in the ensemble.

391 The reader may notice that if  $\xi_i \leq \xi_j$  then  $\bar{R}_{(\xi_j)}(x) \subseteq \bar{R}_{(\xi_i)}(x)$ , which could  
 392 produce correlated base classifiers. In order to increase diversity we can in-  
 393 troduce randomization. Instance-based learners depend mostly on attributes  
 394 used to compute the distance/similarity between objects. This suggests that  
 395 we can promote diversity by using a random subset of attributes (i.e., ran-  
 396 dom subspace method). However, in the context of rough classification, using  
 397 random attributes may not be the best choice since rough approximations  
 398 are often built upon a *reduct* of the attribute set.

399 Besides, unlike nearest-neighbor classifiers, RCN-based models are sensi-  
 400 tive to perturbations on the training data due to the presence of the simi-  
 401 larity threshold, even when using the same subset of attributes. Hence, we  
 402 can perform *instance bagging* [51] in order to counter the correlation coming  
 403 from the rule  $\xi_i \leq \xi_j \implies \bar{R}_{(\xi_j)}(x) \subseteq \bar{R}_{(\xi_i)}(x)$ . Instance bagging attempts  
 404 to neutralize the instability of the base classifier by modifying the original

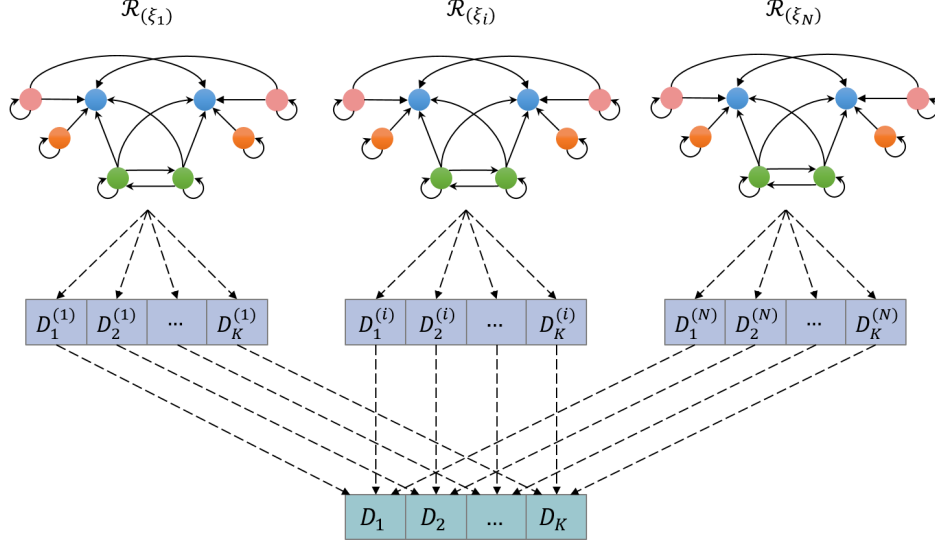


Figure 2: Rough Cognitive Ensemble of  $N$  networks for problems with  $M$  classes.

405 training set (i.e., deleting some instances and replicating others). During this  
 406 process, instances are randomly sampled with replacement from the original  
 407 dataset to create a new one with the same size [52]. This allows establishing  
 408 a reasonable trade-off between ensemble diversity and accuracy.

#### 409 4.2. The exploitation scheme

410 The exploitation phase of the proposed granular ensemble is focused on  
 411 computing a vector  $\mathcal{A}_x(\mathcal{D}^{(i)}) = [A_x(D_1^{(i)}), \dots, A_x(D_k^{(i)}), \dots, A_x(D_K^{(i)})]$  for each  
 412 granular classifier. In this output vector,  $A_x(D_k^{(i)})$  represents the activation  
 413 degree of the  $k$ th output neuron according to the  $i$ th granular network, with  
 414  $x$  being the test instance. This allows computing the aggregated vector  
 415  $\mathcal{A}_x(\mathcal{D}) = [A_x(D_1), \dots, A_x(D_k), \dots, A_x(D_K)]$  by combining the responses of  
 416 the  $N$  base granular classifiers over  $K$  decision classes.

417 In order to activate the ensemble, we need to compute  $N$  excitation vec-  
 418 tors  $\{\mathcal{A}_{[x|\xi_i]}^{(0)}\}_{i=1}^N$  where  $\mathcal{A}_{[x|\xi_i]}^{(0)}$  is used to perform the neural reasoning process  
 419 in the  $i$ th cognitive network. The  $i$ th activation vector denotes the inclusion  
 420 degree of the similarity class  $\bar{R}_{(\xi_i)}(x)$  into each information granule, accord-  
 421 ing to the corresponding similarity threshold  $\xi_i$ . Formally, the  $i$ th activation  
 422 vector could be mathematically defined as follows:

$$\begin{aligned} \mathcal{A}_{[x|\xi_i]}^{(0)} = & [A_{[x|\xi_i]}^{(0)}(P_1), \dots, A_{[x|\xi_i]}^{(0)}(P_k), \dots, A_{[x|\xi_i]}^{(0)}(P_K), \\ & \dots, A_{[x|\xi_i]}^{(0)}(N_1), \dots, A_{[x|\xi_i]}^{(0)}(N_k), \dots, A_{[x|\xi_i]}^{(0)}(N_K), \dots, \\ & A_{[x|\xi_i]}^{(0)}(B_1), \dots, A_{[x|\xi_i]}^{(0)}(B_k), \dots, A_{[x|\xi_i]}^{(0)}(B_K)] \end{aligned}$$

423 where  $A_{[x|\xi_i]}^{(0)}(P_k)$ ,  $A_{[x|\xi_i]}^{(0)}(N_k)$  and  $A_{[x|\xi_i]}^{(0)}(B_k)$  denote the activation degree  
 424 of the  $k$ th positive, negative and boundary region, respectively. These values  
 425 are computed according to the activation rules  $R_6 - R_8$ .

426 When activating the granular ensemble classifier, output-type neurons  
 427 remain inactive since their activation value is computed from the recurrent  
 428 propagation of the evidence over the  $i$ th granular network. This implies that  
 429  $A_{[x|\xi_i]}^{(0)}(D_1) = \dots = A_{[x|\xi_i]}^{(0)}(D_k) = \dots = A_{[x|\xi_i]}^{(0)}(D_K) = 0$ . Sometimes, there is  
 430 additional knowledge about the probability of producing a specific decision  
 431 class given an instance (e.g., resulting from an intermediate classification  
 432 process). In these cases, we could use this knowledge to activate the output  
 433 neurons in order to improve the performance.

434 Once the neural reasoning step is completed (i.e., either an equilibrium  
 435 point is discovered or a maximal number of iterations is reached), the pre-  
 436 dicted class is derived from the aggregated output vector:

$$\mathcal{A}_x(D) = [A_x(D_1), \dots, A_x(D_k), \dots, A_x(D_K)].$$

437 This aggregated vector is obtained by voting on all  $N$  base classifiers over  
 438 the  $K$  decision classes. Therefore, the activation degree  $A_x(D_k)$  denotes the  
 439 probability  $P(D_k|x \in X)$  of producing the  $k$ th decision class given  $x \in X$ ,  
 440 and is computed according to Equation (2). The reader may notice that we  
 441 can simulate the voting scheme in the cognitive network by connecting the  
 442  $k$ th output-type neuron  $D_k^{(i)}$  with the corresponding decision neuron, and  
 443 next performing a single-ahead reasoning process.

444 Sometimes we could identify scenarios on which multiple decision classes  
 445 have the highest probability of being produced (i.e., two aggregated neurons  
 446 bear the same maximal activation value). These situations are common in  
 447 highly inconsistent problems where there is a lack of conclusive knowledge.  
 448 To solve this issue, we select the decision class  $D_k$  associated with the closest  
 449 neighbor of the test instance  $x \in X$ , such that  $D_k$  has the highest output  
 450 probability. This heuristic provides a fair compromise between the decision



451 class computed by the granular ensemble and the intrinsic relation among  
 452 those instances that represent the same pattern.

## 453 5. Heterogeneous distance functions

454 As mentioned, the distance function plays a pivotal role when designing  
 455 instance-based classifiers. As far as we know, there is no study on the impact  
 456 of this component on the performance of RCN-based classifiers. In this sec-  
 457 tion, we revise the mathematical formulation of three widely used distance  
 458 functions taken from [53] that allow comparing heterogeneous instances, i.e.,  
 459 objects comprising both numerical and nominal attributes.

460 Let  $\Phi = \{\phi_1, \dots, \phi_M\}$  denote the attribute set, where  $\phi_j$  can be either  
 461 numerical or nominal, and it has a weight  $0 \leq \omega_j \leq 1$  attached that quantifies  
 462 its relevance. The dissimilarity function  $\delta(x, y)$  between two instances  $x$  and  
 463  $y$  can be computed using one of the following distance functions:

- 464 • **The Heterogeneous Euclidean-Overlap Metric (HEOM).** This  
 465 heterogeneous distance function computes the normalized Euclidean  
 466 distance between numerical attributes and an overlap metric for nomi-  
 467 nal attributes. Equations (3) and (4) define the HEOM distance func-  
 468 tion, where  $x(j)$  and  $y(j)$  are the normalized values of the  $j$ th attribute  
 469 for heterogeneous instances  $x$  and  $y$ , respectively.

$$\delta_{HEOM}(x, y) = \sqrt{\frac{\sum_{j=1}^{|\Phi|} \omega_j \sigma_j(x, y)}{\sum_{j=1}^{|\Phi|} \omega_j}} \quad (3)$$

470 where

$$\sigma_j(x, y) = \begin{cases} 0 & \text{if } \phi_j \text{ is nominal } \wedge x(j) = y(j) \\ 1 & \text{if } \phi_j \text{ is nominal } \wedge x(j) \neq y(j) \\ (x(j) - y(j))^2 & \text{if } \phi_j \text{ is numerical} \end{cases} \quad (4)$$

471

- 472 • **The Heterogeneous Manhattan-Overlap Metric (HMOM).** This  
 473 heterogeneous variant is similar to the HEOM function since it replaces  
 474 the Euclidean distance with the Manhattan distance when computing

475 the dissimilarity between two numerical values. Equations (5) and (6)  
 476 formalize the HMOM distance function, whose calculation requires less  
 477 computational effort compared to HEOM's.

$$\delta_{HMOM}(x, y) = \frac{\sum_{j=1}^{|\Phi|} \omega_j \rho_j(x, y)}{\sum_{j=1}^{|\Phi|} \omega_j} \quad (5)$$

478 where

$$\rho_j(x, y) = \begin{cases} 0 & \text{if } \phi_j \text{ is nominal} \wedge x(j) = y(j) \\ 1 & \text{if } \phi_j \text{ is nominal} \wedge x(j) \neq y(j) \\ |x(j) - y(j)| & \text{if } \phi_j \text{ is numerical} \end{cases} \quad (6)$$

479

480 • **The Heterogeneous Value Difference Metric (HVDM)**. This  
 481 function involves a stronger strategy for quantifying the dissimilarity  
 482 between two discrete attribute values. Instead of computing the match-  
 483 ing between attribute values, it measures the correlation between such  
 484 attributes and decision classes. Equations (7) and (8) show the HVDM  
 485 function variant adopted in this research.

$$\delta_{HVDM}(x, y) = \sqrt{\frac{\sum_{j=1}^{|\Phi|} \omega_j \tau_j(x, y)}{\sum_{j=1}^{|\Phi|} \omega_j}} \quad (7)$$

486 where

$$\tau_j(x, y) = \begin{cases} \frac{1}{K} \sum_{k=1}^K \left( \frac{\beta_{\phi_j, x(j), k}}{\beta_{\phi_j, x(j)}} - \frac{\beta_{\phi_j, y(j), k}}{\beta_{\phi_j, y(j)}} \right)^2 & \text{if } \phi_j \text{ is nominal} \\ (x(j) - y(j))^2 & \text{if } \phi_j \text{ is numerical} \end{cases} \quad (8)$$

487 whereas  $\beta_{\phi_j, x(j)}$  is the number of instances in the training set that have  
 488 value  $x(j)$  for attribute  $\phi_j$ , and  $\beta_{\phi_j, x(j), k}$  denotes the number of instances that  
 489 have value  $x(j)$  for attribute  $\phi_j$  and output class  $k$ .

490 In these distance functions, approximating the relevance of each attribute  
 491 may result in improved prediction rates. To overcome this issue, we compute  
 492 the *gain ratio* [54] associated with the  $j$ th attribute:

$$G(\phi_j, X) = \frac{I(X) - E(\phi_j, X)}{IC(\phi_j)} \quad (9)$$

493 where

$$I(X) = - \sum_{k=1}^K \frac{|X \cap D_k|}{|X|} \log_2 \frac{|X \cap D_k|}{|X|} \quad (10)$$

494 measures the randomness of the distribution of available instances in  $X$  over  
 495  $K$  decision classes, whereas  $E(\phi_j, X)$  is given by

$$E(\phi_j, X) = \sum_{l=1}^{L_j} \frac{|X_l|}{|X|} I(X_l) \quad (11)$$

496 where  $L_j$  is the number of possible values for the  $\phi_j$  attribute and  $X_l$  denotes  
 497 the instance set in  $X$  having value  $v_l$  for the  $\phi_j$  attribute. In order to counter  
 498 this metric's bias in favor of attributes with a larger number of values, the  
 499 following normalization factor is introduced:

$$IC(\phi_j) = - \sum_{l=1}^{L_j} \frac{|X_l|}{|X|} \log_2 \frac{|X_l|}{|X|} \quad (12)$$

500 In our approach, the gain ration associated to the  $j$ th attribute replaces  
 501 the weight  $\omega_j$  when computing the dissimilarity degree between two hetero-  
 502 geneous instances. If  $\sum \omega_j = 0$ , then attribute weighting is not performed.  
 503 Of course, we can adopt more sophisticated measures as the distance between  
 504 partitions, but it will increase the computational cost.

## 505 6. Numerical simulations

506 In this section, we conduct several experiments in order to evaluate the  
 507 prediction capabilities of the proposed ensemble classifier. With this goal in  
 508 mind, we first determine the best-performing variant among multiple ensemble  
 509 models and demonstrate RCE's superiority with regards to the RCN-based  
 510 classifiers. In these simulations we also evaluate the algorithm's performance  
 511 for different distance functions. Afterwards, we compare the prediction ca-  
 512 pability of the best-performing ensemble method against 15 well-established  
 513 classifiers across 140 pattern classification datasets.

514 *6.1. Dataset characterization*

515 We leaned upon 140 well-known pattern classification datasets taken from  
 516 the KEEL [55] and UCI ML [56] repositories. These ML problems comprise  
 517 different characteristics and allow evaluating the predictive capability of both  
 518 state-of-the-art and granular algorithms under consideration.

519 Table 1 outlines the number of instances, attributes and decision classes  
 520 for each dataset. The presence of noise and the imbalance ratio (calculated  
 521 as the ratio of the size of the majority class to that of the minority class) are  
 522 also given. In this paper, we say that a dataset is imbalanced if the number  
 523 of instances belonging to the majority decision class is at least five times the  
 524 number of instances belonging to the minority class.

Table 1: Characterization of the ML datasets adopted for the simulations.

Dataset	Instances	Attributes	Classes	Noisy	Imbalance
abalone	4174	8	28	no	689:1
acute-inflammation	120	6	2	no	no
acute-nephritis	120	6	2	no	no
anneal	898	38	6	no	85:1
anneal.orig	898	38	6	no	85:1
appendicitis	106	7	2	no	no
arrhythmia	452	262	13	no	122:1
audiology	226	69	24	no	57:1
australian	690	14	2	no	no
autos	205	25	7	no	22:1
balance-noise	625	4	3	yes	5:1
balance-scale	625	4	3	no	5:1
ballons	16	4	2	no	no
banana	5300	2	2	no	no
bank	4521	16	2	no	7:1
blood	748	4	2	no	no
breast	277	9	2	no	no
bc-wisconsin-diag	569	31	2	no	no
bc-wisconsin-prog	198	34	2	no	no
bridges-version1	107	12	6	no	no
bridges-version2	107	12	6	no	no
car	1728	6	4	no	17:1

*Continued on next page*

Table 1 – *Continued from previous page*

Dataset	Instances	Attributes	Classes	Noisy	Imbalance
cardiotocography-10	2126	35	10	no	10:1
cardiotocography-3	2126	35	3	no	9:1
chess	3196	36	2	no	no
cleveland	297	13	5	no	12:1
colic	368	22	2	no	no
colic.orig	368	27	2	no	no
collins	500	23	15	no	13:1
contact-lenses	24	4	3	no	no
contraceptive	1473	9	3	no	no
credit-a	690	15	2	no	no
credit-g	1000	20	2	no	no
crx	653	15	2	no	no
csj	653	34	6	no	no
cylinder-bands	540	39	2	no	no
dermatology	358	34	6	no	5:1
echocardiogram	131	11	2	no	5:1
ecoli	336	7	8	no	71:1
ecoli0	220	7	2	no	no
ecoli-0vs1	220	7	2	no	no
ecoli1	336	7	2	no	no
ecoli2	336	7	2	no	5:1
ecoli3	336	7	2	no	8:1
ecoli-5an-nn	336	7	8	yes	71:1
energy-y1	768	8	38	no	no
energy-y2	768	8	38	no	no
eucalyptus	736	19	5	no	no
flags	194	28	8	no	15:1
glass	214	9	6	no	8:1
glass0	214	9	2	no	no
glass-0123vs456	214	9	2	no	no
glass1	214	9	2	no	no
glass-10an-nn	214	9	6	yes	8:1
glass2	214	9	2	no	no
glass-20an-nn	214	9	6	yes	8:1
glass3	214	9	2	no	6:1

*Continued on next page*

Table 1 – *Continued from previous page*

Dataset	Instances	Attributes	Classes	Noisy	Imbalance
glass-5an-nn	214	9	6	yes	8:1
glass6	214	9	2	no	6:1
haberman	306	3	2	no	no
hayes-roth	160	4	3	no	no
heart-5an-nn	270	13	2	yes	no
heart-statlog	270	13	2	no	no
hypothyroid	3772	29	4	no	1740:1
ionosphere	351	34	2	no	no
iris	150	4	3	no	no
iris0	150	4	2	no	no
iris-10an-nn	150	4	3	yes	no
iris-20an-nn	150	4	3	yes	no
iris-5an-nn	150	4	3	yes	no
labor	57	16	2	no	no
led7digit	500	7	10	no	no
libras	360	90	15	no	no
liver-disorders	345	6	2	no	no
lung-cancer	32	56	3	no	no
lymph	148	18	4	no	40:1
mammographic	830	5	2	no	no
mfeat-factors	2000	216	10	no	no
mfeat-fourier	2000	76	10	no	no
mfeat-karhunen	2000	64	10	no	no
mfeat-morpho	2000	6	10	no	no
mfeat-pixel	2000	240	10	no	no
mfeat-zernike	2000	47	10	no	no
molecular-biology	106	57	2	no	no
monk-2	432	6	2	no	no
mushroom	5644	22	2	no	no
musk-1	476	167	2	no	no
musk-2	6598	167	2	no	5:1
new-thyroid	215	5	2	no	5:1
nursery	12960	8	5	no	2160:1
optdigits	5620	64	10	no	no
ozone	2536	72	2	no	33:1

*Continued on next page*

Table 1 – *Continued from previous page*

Dataset	Instances	Attributes	Classes	Noisy	Imbalance
page-blocks	5473	10	5	no	175:1
parkinsons	195	22	2	no	no
pendigits	10992	16	10	no	no
phoneme	5404	5	2	no	no
pima	768	8	2	no	no
pima-10an-nn	768	8	2	yes	no
pima-20an-nn	768	8	2	yes	no
pima-5an-nn	768	8	2	yes	no
planning	182	12	2	no	no
plant-margin	1600	64	100	no	no
plant-shape	1600	64	100	no	no
plant-texture	1599	64	100	no	no
postoperative	90	8	3	no	32:1
primary-tumor	339	17	22	no	84:1
saheart	462	9	2	no	no
segment	2310	19	7	no	no
solar-flare-1	323	5	6	no	11:1
solar-flare-2	1066	12	6	no	7:1
sonar	208	60	2	no	no
soybean	683	35	19	no	11:1
spambase	4601	57	2	no	no
spectfheart	267	44	2	no	no
spectrometer	531	101	48	no	29:1
splice	3190	60	3	no	no
sponge	76	44	3	no	23:1
tae	151	5	3	no	no
tic-tac-toe	958	9	2	no	no
vehicle	846	18	4	no	no
vehicle0	846	18	2	no	no
vehicle1	846	18	2	no	no
vehicle2	846	18	2	no	no
vehicle3	846	18	2	no	no
vertebral2	310	6	2	no	no
vertebral3	310	6	3	no	no
vote	435	16	2	no	no

*Continued on next page*

Table 1 – *Continued from previous page*

Dataset	Instances	Attributes	Classes	Noisy	Imbalance
vowel	990	13	11	no	no
wall-following	5456	24	4	no	6:1
waveform	5000	40	3	no	no
weather	14	4	2	no	no
wine	178	13	3	no	no
wine-5an-nn	178	13	3	yes	no
winequality-red	1599	11	6	no	68:1
winequality-white	4898	11	7	no	439:1
wisconsin	683	9	2	no	no
yeast	1484	8	10	no	92:1
yeast1	1484	8	2	no	no
yeast3	1484	8	2	no	8:1
zoo	101	16	7	no	10:1

525

526 In the adopted datasets, the number of attributes ranges from 2 to 262,  
527 the number of decision classes from 2 to 100, and the number of instances  
528 from 14 to 12,906. These ML problems involve 13 noisy and 47 imbalanced  
529 datasets, where the imbalance ratio ranges from 5:1 to 2160:1. In order to  
530 avoid the out-of-range issues in the adopted heterogeneous distance func-  
531 tions, the numerical attributes have been normalized. On the other hand, we  
532 replaced missing values by the mean or the mode depending if the attribute  
533 is numerical or nominal, respectively.

534 Each dataset has been partitioned using a 10-fold cross-validation proce-  
535 dure, i.e., the dataset has been split into ten folds, each one containing ap-  
536 proximately 10% of the instances. For each fold, a ML algorithm is trained  
537 with the instances contained in the training partition (9 folds) and then  
538 tested with the current fold. It should be mentioned that test partitions are  
539 kept aside to evaluate the performance of the learned model.

#### 540 6.2. State-of-the-art classifiers used for comparison

541 In this subsection, we outline the classifiers adopted for benchmarking  
542 purposes. The variants and default parameter settings implemented in Weka  
543 v3.6.11 [57] have been retained throughout the simulations. It should be high-  
544 lighted that these default parameter values are common across all datasets,  
545 thus no algorithm performs hyperparameter tuning.



546 As Triguero et. al. [58] stated, a good choice of parameter values increases  
547 the algorithms' performance over different data sources. Nevertheless, a ro-  
548 bust model should allow good-enough results to be obtained even when its  
549 parameters might not have been optimized for a specific dataset. Observe  
550 that robustness is not a requirement for accuracy since we can optimize the  
551 model parameters and produce higher prediction rates, but it also increases  
552 the computational complexity of setting up the classifier.

553 In spite of the above facts, the key reason behind the decision of not per-  
554 forming hyperparameter learning relies on the aim of our research, namely:  
555 *to suppress the parametric requirements when constructing a Rough Cognitive*  
556 *Network*. Accordingly, the following simulations focus on examining the pre-  
557 diction capability of the ensemble model against well-established classifiers,  
558 even when no algorithm undergoes parameter tuning.

559 The classifiers used for comparison are: Decision Table (DT) [59], Naive  
560 Bayes (NB) [60], Naive Bayes Updateable (NBU) [60], Support Vector Ma-  
561 chines (SMO) with sequential minimal optimization algorithm [61], Multi-  
562 layer Perceptron (MLP) [62], Simple Logistic (SL) [63], Decision Tree (J48)  
563 [64], Fast Decision Trees (FDT) [65], Best-first Decision Trees (BFT) [66],  
564 Logistic Model Trees (LMT) [67], Random Trees (RT) [68], Random Forests  
565 (RF) [18],  $k$ -nearest neighbors learner (kNN) [69],  $K^*$  instance-based classi-  
566 fier ( $K^*$ ) [70] and Locally Weighted Learning (LWL) [71].

567 In the case of RCN variants, we set the similarity threshold to  $\xi = 0.98$   
568 for all datasets while the similarity threshold  $\xi_i$  associated with the  $i$ th base  
569 classifier is uniformly distributed in the  $[0.96, 1)$  interval. Aiming at reducing  
570 the complexity of the granular ensemble, we employed  $N = 10$  base classifiers  
571 in the subsequent numerical simulations.

### 572 6.3. Determining the best-performing granular classifier

573 The first simulation focused on determining the best-performing granular  
574 model. With this goal in mind, we evaluated the prediction capability of  
575 standard RCNs, the ensemble model without bagging (RCE) and the ensemble  
576 model performing instance bagging (RCB) across three heterogeneous  
577 distance functions. Figure 3 displays the average Cohen's kappa coefficient  
578 achieved by each ML algorithm. The Cohen's kappa coefficient [72] measures  
579 the inter-rater agreement for categorical items. It is usually deemed a more  
580 robust measure than the standard accuracy since this coefficient takes into  
581 account the agreement occurring by chance.

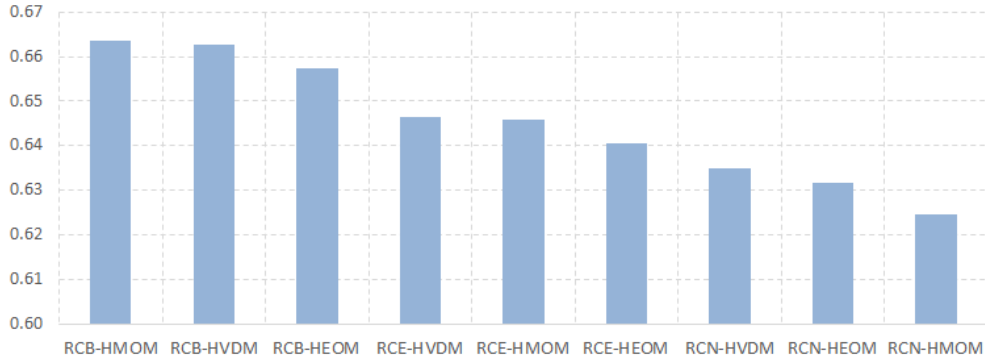


Figure 3: Average Kappa measure computed by the rough classifiers.

582 In order to examine the existence of statistically significant differences  
 583 in performance, we computed the Friedman two-way analysis of variances  
 584 by ranks [73]. The Friedman test is a multiple-comparison nonparametric  
 585 statistical method that detects whether at least two of the samples in a  
 586 group represent populations with different median values or not. The test  
 587 suggests rejecting the null hypothesis  $H_0$  ( $p$ -value =  $7.989819E - 11 < 0.05$ )  
 588 using a confidence interval of 95%. This implies that there exist statistically  
 589 significant performance differences between at least two algorithms across  
 590 all the selected datasets. Figure 4 shows the rank values computed by the  
 591 Friedman test, where the RCB algorithm using the HVDM distance function  
 592 emerged as the best-ranked granular classifier.

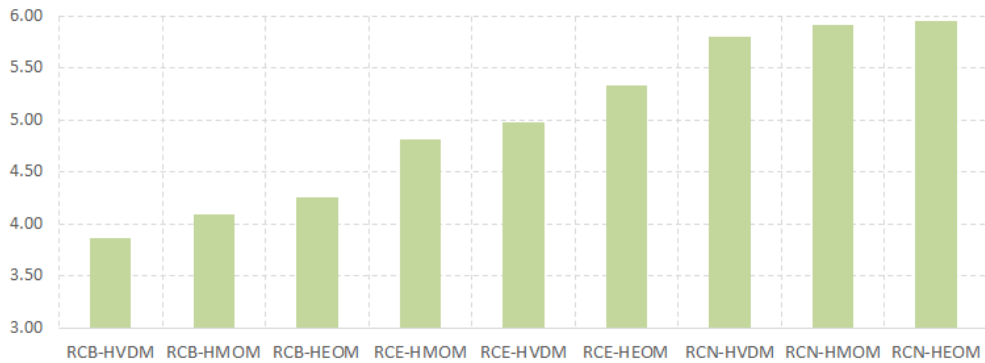


Figure 4: Friedman's rank for the rough classifiers.

593 The next step is to determine whether the superiority of the RCB-HVDM  
 594 classifier is statistically significant or not. By doing so, we resorted to the  
 595 Wilcoxon signed rank test [74] and several post-hoc procedures to adjust the  
 596  $p$ -values, as recently suggested Benavoli et al. [75]. The post-hoc procedures  
 597 are required since in pairwise analysis, if we try to draw a conclusion involving  
 598 more than one pairwise comparison, we accumulate an error coming from  
 599 their combination. Therefore, we are losing control on the *Family-Wise Error*  
 600 *Rate*, defined as the probability of making one or more false discoveries among  
 601 all the hypotheses associated to multiple pairwise tests.

602 Table 2 reports the unadjusted  $p$ -value computed by the Wilcoxon signed  
 603 rank test as well as the corrected  $p$ -values associated with each pairwise com-  
 604 parison using RCB-HVDM as the control method. We assume that a null  
 605 hypothesis  $H_0$  can be rejected if at least one of the adopted post-hoc proce-  
 606 dures supports the rejection. The statistical analysis supports the superiority  
 607 of the RCB-HVDM classifier as all the conservative hypotheses were rejected,  
 608 save for the one concerning to the RCB-HMOM vs. RCB-HVDM pair. This  
 609 suggests that RCB-HMOM and RCB-HVDM perform comparably although  
 610 the RCB-HMOM algorithm is ranked first.

Table 2: Adjusted  $p$ -values according to different post-hoc procedures using the best-  
 performing ensemble classifier (RCB-HVDM) as a control method.

Algorithm	$p$ -value	Bonferroni	Holm	Holland	Hypothesis
RCN-HEOM	6.67E-10	5.33E-09	5.33E-09	5.33E-09	Rejected
RCN-HVDM	3.44E-08	2.75E-07	2.40E-07	2.40E-07	Rejected
RCN-HMOM	5.59E-08	4.47E-07	3.35E-07	3.35E-07	Rejected
RCE-HEOM	1.75E-06	1.40E-05	8.76E-06	8.76E-06	Rejected
RCE-HVDM	0.001107	0.008861	0.004430	0.004423	Rejected
RCE-HMOM	0.003157	0.025261	0.009473	0.009443	Rejected
RCB-HEOM	0.010440	0.083525	0.020881	0.020772	Rejected
RCB-HMOM	0.848561	1.000000	0.848561	0.848561	Accepted

611 Figure 5 displays the impact of using (a) different heterogeneous distance  
 612 functions and (b) different ensemble strategies. From this figure we can for-  
 613 malize two key conclusions. First, performing instance bagging is convenient  
 614 regardless of the underlying distance function. Second, it seems that selecting  
 615 an ensemble scheme may reduce the negative effects of using a non-optimal  
 616 distance function for a particular dataset.

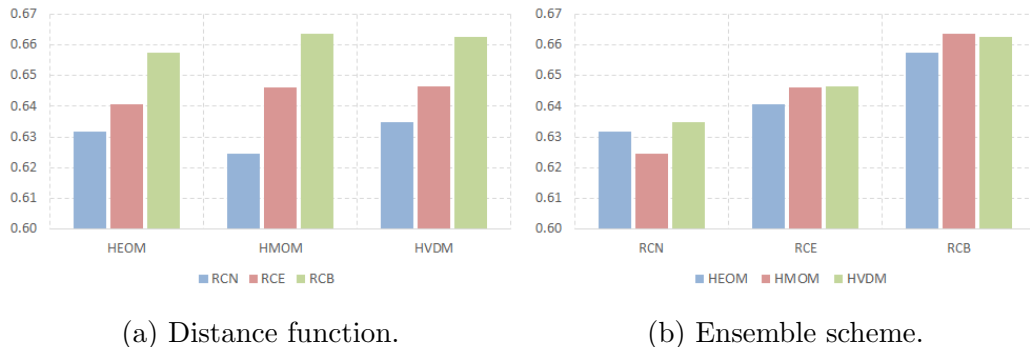


Figure 5: Average Kappa measure according to different criteria.

617 The above simulations confirm the superiority of the RCE-based models  
 618 over the base RCN classifier. Furthermore, the proposed ensemble classifier  
 619 removed the need for an RCN parameter tuning stage, which became one of  
 620 the main motivations of this study.

#### 621 6.4. Comparison against state-of-the-art classifiers

622 In this subsection, we compare the prediction ability of the best-performing  
 623 granular ensemble (RCB-HVDM, hereinafter simply called RCE) against the  
 624 traditional classifiers mentioned in Section 6.2. Analogously to the previous  
 625 simulations, we utilized Cohen’s kappa coefficient to quantify the algorithms’  
 626 performance. Figure 6 portrays the average Kappa measure attained by each  
 627 algorithm across the 140 problems in our testbed.

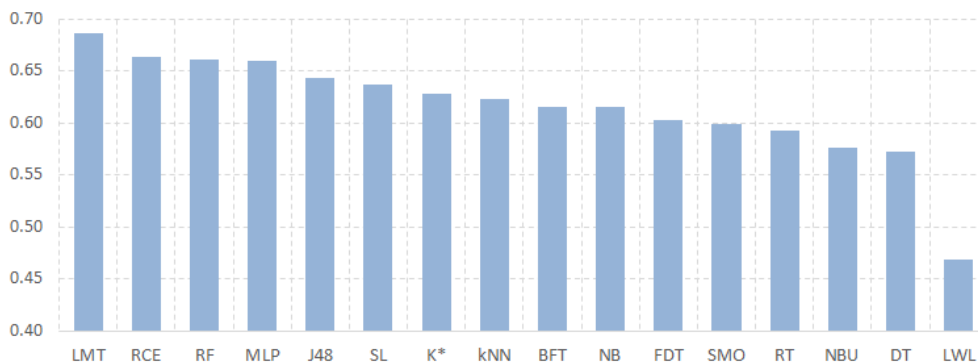


Figure 6: Average Kappa measure computed by adopted classifiers.

628 For this experiment, the Friedman test suggests rejecting the null hypoth-  
 629 esis ( $p\text{-value} = 1.519139E - 10 < 0.05$ ) using a confidence level of 95%, thus  
 630 there are significant differences between at least two algorithms across the  
 631 selected datasets. Figure 7 shows the rank values computed by the Friedman  
 632 test. From these results, it is clear that LMT is the best-ranked algorithm,  
 633 RCE is the fourth-best ranked, whereas LWL is the worst one. LMT refers to  
 634 logistic model trees, i.e., classification trees with logistic regression functions  
 635 at the leaves while LWL is an instance-based classifier that employs locally  
 636 weighted linear regression to make predictions.

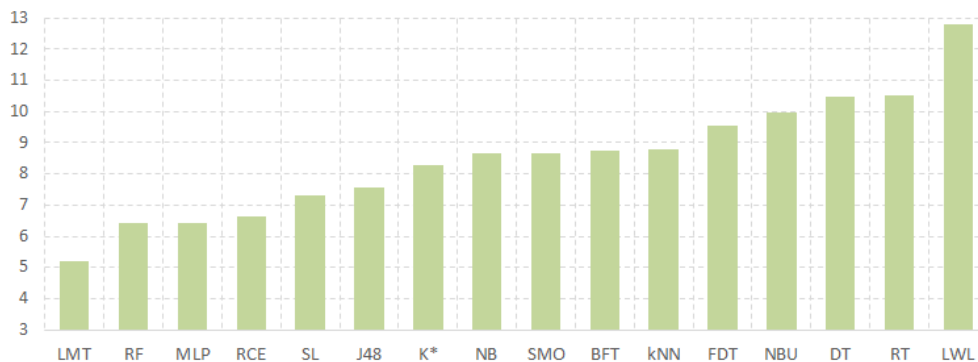


Figure 7: Friedman’s ranking for adopted classifiers.

637 Table 3 shows the  $p$ -values reported by the Wilcoxon signed rank test  
 638 and the corrected  $p$ -values according to the post-hoc procedures using the  
 639 granular ensemble (RCE) as the control method. The results point to the  
 640 fact that LML stands as the best-performing classifier in our study, with no  
 641 significant differences spotted between our proposal and RF, MLP, SL and  
 642 J48, as the null hypothesis was accepted in each of these pairwise compar-  
 643 isons. Note however that SL and J48 report slightly lower Kappa measures.  
 644 More importantly, the granular ensemble is capable of outperforming the re-  
 645 maining classifiers. This confirms the reliability of the granular ensemble in  
 646 solving ML problems with a wide range of features.

647 The last experiment is oriented to characterizing those problems for which  
 648 RCEs constitute the best algorithmic choice. Nevertheless, detecting these  
 649 features manually from a table comprising 140 datasets and 15 algorithms is  
 650 quite challenging and may lead to misleading conclusions.

Table 3: Adjusted  $p$ -values according to different post-hoc procedures using the proposed ensemble classifier (RCB-HVDM) as a control method.

Algorithm	$p$ -value	Bonferroni	Holm	Holland	Hypothesis
LWL	3.22E-19	4.83E-18	4.83E-18	0.000000	Rejected
RT	3.11E-12	4.66E-11	4.35E-11	4.35E-11	Rejected
DT	6.67E-10	1.00E-08	8.67E-09	8.67E-09	Rejected
NBU	8.75E-08	1.31E-06	1.05E-06	1.05E-06	Rejected
kNN	4.47E-07	6.70E-06	4.91E-06	4.91E-06	Rejected
FDT	1.52E-06	2.28E-05	1.52E-05	1.52E-05	Rejected
NB	8.06E-05	0.001210	7.26E-04	7.25E-04	Rejected
SMO	0.000469	0.007038	0.003753	0.003747	Rejected
K*	0.000471	0.007072	0.003753	0.003747	Rejected
BFT	0.000510	0.007649	0.003753	0.003747	Rejected
LMT	0.000997	0.014962	0.004987	0.004977	Rejected
J48	0.059231	0.888460	0.236922	0.216691	Accepted
RF	0.399342	1.000000	1.000000	0.783288	Accepted
SL	0.486293	1.000000	1.000000	0.783288	Accepted
MLP	0.990836	1.000000	1.000000	0.990835	Accepted

651 *Metalearning* is a subfield of ML where automatic learning algorithms  
652 are applied on metadata concerning classification/prediction simulations. Al-  
653 though different researchers hold different views as to what the term exactly  
654 means, the main goal is to use such metadata (i.e., problems' features) to un-  
655 derstand how learning models can become flexible in solving different kinds  
656 of classification problems [76]. Moreover, under the proper conditions, meta-  
657 learning also allows detecting the features on which a specific algorithm may  
658 outperform a set of baseline classification models.

659 In some scenarios, it is enough to predict whether the algorithm under  
660 analysis will be successful or not, while in others the purpose is to understand  
661 why the algorithm behaves the way it does.

662 In this paper, we use the well-known  $k$ NN classifier [69] as baseline algo-  
663 rithm since both belong to the same family (i.e., instance-based classifiers).  
664 The ML metadata include five features, namely: the number of instances,  
665 number of attributes, number of classes, whether the problem is noisy or not,  
666 and the imbalance ratio. Moreover, we consider a decision attribute with two  
667 classes ( $P$  - *positive*,  $N$  - *negative*). Observe that this dataset will comprise  
668 the algorithms' behaviors for adopted problems.

669 Therefore, a ML problem will be labeled as *positive* if the proposed algo-  
 670 rithm was capable of outperforming the baseline classifier, otherwise it will  
 671 be labeled as *negative*. More explicitly, we assume that an instance is *positive*  
 672 if  $\kappa(\text{RCE}) - \kappa(k\text{NN}) \geq 0.05$ , with  $\kappa(\cdot)$  denoting the kappa coefficient for a  
 673 particular problem. If  $\kappa(\text{RCE}) - \kappa(k\text{NN}) \leq -0.05$  then the instance is labeled  
 674 as *negative*, while those problems where  $-0.05 < |\kappa(\text{RCE}) - \kappa(k\text{NN})| < 0.05$   
 675 were not included in the new dataset in order to increase the separability  
 676 between instances used to build the metaclassifier.

677 For instance, the *audiology* problem includes 226 objects, 69 attributes, 24  
 678 decision classes and it does not involve noisy features, but a high imbalance  
 679 degree (57:1). Since  $\kappa(\text{RCE}) = 0.75$  and  $\kappa(k\text{NN}) = 0.59$  we can infer the  
 680 synthetic instance: “226, 69, 24, no, high, P”. Following the same procedure,  
 681 we obtain an artificial ML dataset comprising 63 instances (i.e., 47 labeled  
 682 as positive and 16 labeled as negative). From this knowledge we can identify  
 683 when our classification model is the best choice.

684 The next step focuses on training a decision tree classifier [64] to predict  
 685 whether the ensemble will be adequate for solving a new ML problem or not.  
 686 More importantly, decision tree classifiers allow deriving rules to interpret  
 687 the prediction results. Figure 8 displays the unpruned decision tree obtained  
 688 after running the underlying learning scheme.

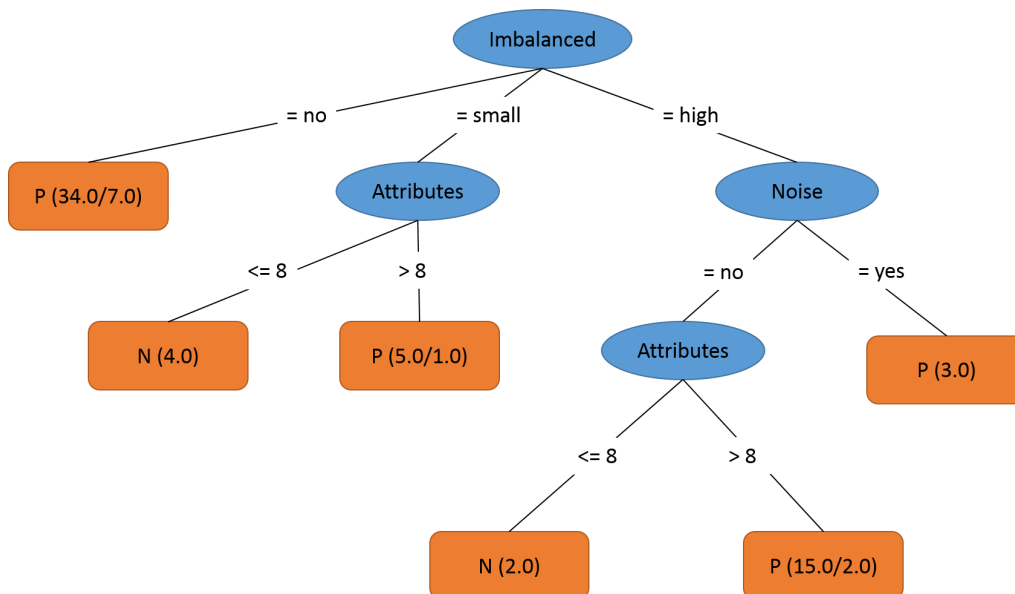


Figure 8: Decision tree characterizing the ensemble performance.

689 The decision tree classifier yielded an 86% accuracy, which can be in-  
690 terpreted as the confidence of the derived rule set. Such rules suggest that  
691 the granular ensemble outperforms the  $k$ NN algorithm in rather balanced  
692 problems, or alternatively in those problems that exhibit a certain imbal-  
693 ance degree but are defined by a reasonable number of attributes. Moreover,  
694 the derived rules have shown that our classifier seems to be less sensitive to  
695 noise, which could be a direct result of performing instance bagging over base  
696 classifiers with variable granularity levels.

## 697 7. Concluding remarks

698 In this paper, we introduced the notion of *Rough Cognitive Ensembles* as  
699 an alternative to deal with the hyperparameter learning requirements of the  
700 RCN algorithm. This granular classifier can be defined as a multiclassifier  
701 system composed of a set of RCNs, each operating at a different granularity  
702 degree. It should be noticed that inducing different granularity degrees can  
703 be achieved by using a different similarity threshold for each base classifier.  
704 On the other hand, we perform instance bagging to reduce the correlation  
705 coming from the rule  $\xi_i \leq \xi_j \implies \bar{R}_{(\xi_j)}(x) \subseteq \bar{R}_{(\xi_i)}(x)$ .

706 We conducted an extensive empirical analysis to assess the prediction  
707 capability of the proposed classifier. The results have corroborated that the  
708 ensemble model performing instance bagging leads to higher prediction rates  
709 regardless of the underlying dissimilarity functional. In spite of this fact, the  
710 HVDM function often reports improved prediction rates.

711 From the comparison between the best-performing ensemble model and  
712 15 state-of-the-art classifiers, we can confirm the reliability of our proposal  
713 in solving pattern classification problems. The statistical analysis concluded  
714 that our granular classifier is capable of outperforming most ML algorithms  
715 while remaining competitive w.r.t. RF, MLP, SL and J48.

716 On the other hand, there is no doubt about the statistical superiority of  
717 RCEs when compared to other well-known instance-based learners. Aiming  
718 at explaining this behavior, we derived a rule set from a decision tree using  
719 the  $k$ NN learner as baseline classifier. The obtained rules have shown that the  
720 proposed algorithm performs well in presence of imbalanced classes as long as  
721 the problem comprises a reasonable number of attributes. In spite of these  
722 findings, the future research will be focused on exploring other appealing  
723 theoretical properties behind this novel rough classifier.



724 **Acknowledgements**

725 This work was supported by the Research Council of Hasselt University.  
726 The authors would like to thank the anonymous reviewers for their construc-  
727 tive remarks throughout the revision process.

728 **References**

- 729 [1] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, 2nd Edition,  
730 John Wiley & Sons, 2012.
- 731 [2] Y. Sun, A. K. Wong, M. S. Kamel, Classification of imbalanced data:  
732 a review, *International Journal of Pattern Recognition and Artificial*  
733 *Intelligence* 23 (04) (2009) 687–719.
- 734 [3] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight  
735 into classification with imbalanced data: Empirical results and current  
736 trends on using data intrinsic characteristics, *Information Sciences* 250  
737 (2013) 113–141.
- 738 [4] B. Frénay, M. Verleysen, Classification in the presence of label noise: a  
739 survey, *Neural Networks and Learning Systems, IEEE Transactions on*  
740 25 (5) (2014) 845–869.
- 741 [5] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, T. S. Huang, Semisuper-  
742 vised learning of classifiers: Theory, algorithms, and their application  
743 to human-computer interaction, *Pattern Analysis and Machine Intelli-*  
744 *gence, IEEE Transactions on* 26 (12) (2004) 1553–1566.
- 745 [6] G. Tsoumakas, I. Katakis, Multi-label classification: an overview, *Inter-*  
746 *national Journal of Data Warehousing and Mining* 3 (3) (2007) 1–13.
- 747 [7] W. Cheng, E. Hüllermeier, K. J. Dembczynski, Graded multilabel clas-  
748 sification: the ordinal case, in: *Proceedings of the 27th international*  
749 *conference on machine learning (ICML-10)*, 2010, pp. 223–230.
- 750 [8] G. Nápoles, R. Falcon, E. Papageorgiou, R. Bello, K. Vanhoof, Partitive  
751 granular cognitive maps to graded multilabel classification, in: *Fuzzy*  
752 *Systems (FUZZ-IEEE)*, 2016 IEEE International Conference on, IEEE  
753 Computational Intelligence Society, 2016.

- 754 [9] S. B. Kotsiantis, Decision trees: a recent overview, *Artificial Intelligence*  
755 *Review* 39 (4) (2013) 261–283.
- 756 [10] G. P. Zhang, Neural networks for classification: a survey, *Systems, Man,*  
757 *and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*  
758 *on* 30 (4) (2000) 451–462.
- 759 [11] H. Ishibuchi, T. Nakashima, T. Morisawa, Voting in fuzzy rule-based sys-  
760 *tems for pattern classification problems, Fuzzy sets and systems* 103 (2)  
761 (1999) 223–238.
- 762 [12] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers,  
763 *Machine learning* 29 (2-3) (1997) 131–163.
- 764 [13] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, B. Scholkopf, Support  
765 *vector machines, Intelligent Systems and their Applications, IEEE* 13 (4)  
766 (1998) 18–28.
- 767 [14] T. Cover, P. Hart, Nearest neighbor pattern classification, *IEEE Trans-*  
768 *actions on Information Theory* 13 (1) (1967) 21–27.
- 769 [15] T. G. Dietterich, Ensemble methods in machine learning, in: *Multiple*  
770 *classifier systems, Springer, 2000, pp. 1–15.*
- 771 [16] Y. Ren, L. Zhang, P. Suganthan, Ensemble classification and regression  
772 *- recent developments, applications and future directions, IEEE Com-*  
773 *putational Intelligence Magazine* 11 (1) (2016) 41–53.
- 774 [17] W. Fan, S. J. Stolfo, J. Zhang, The application of adaboost for dis-  
775 *tributed, scalable and on-line learning, in: Proceedings of the Fifth ACM*  
776 *SIGKDD International Conference on Knowledge Discovery and Data*  
777 *Mining, ACM, 1999, pp. 362–366.*
- 778 [18] L. Breiman, Random forests, *Machine learning* 45 (1) (2001) 5–32.
- 779 [19] Y. Zhao, J. Gao, X. Yang, A survey of neural network ensembles, in:  
780 *Neural Networks and Brain, 2005. International Conference on, Vol. 1,*  
781 *IEEE, 2005, pp. 438–442.*
- 782 [20] P. Yang, Y. Hwa Yang, B. B Zhou, A. Y Zomaya, A review of ensemble  
783 *methods in bioinformatics, Current Bioinformatics* 5 (4) (2010) 296–308.

- 784 [21] S. K. Pal, L. Polkowski, *Rough-Neural Computing: Techniques for Com-*  
785 *puting with Words*, Springer-Verlag Berlin Heidelberg, 2004.
- 786 [22] L. Polkowski, P. Artiemjew, *On granular rough computing: factoring*  
787 *classifiers through granulated decision systems*, in: *Rough Sets and In-*  
788 *telligent Systems Paradigms*, Springer, 2007, pp. 280–289.
- 789 [23] W. Pedrycz, B.-J. Park, S.-K. Oh, *The design of granular classifiers:*  
790 *a study in the synergy of interval calculus and fuzzy sets in pattern*  
791 *recognition*, *Pattern Recognition* 41 (12) (2008) 3720–3735.
- 792 [24] R. Al-Hmouz, W. Pedrycz, A. Balamash, A. Morfeq, *From data to gran-*  
793 *ular data and granular classifiers*, in: *Fuzzy Systems (FUZZ-IEEE), 2014*  
794 *IEEE International Conference on*, IEEE, 2014, pp. 432–438.
- 795 [25] R. Al-Hmouz, W. Pedrycz, A. Balamash, A. Morfeq, *Description and*  
796 *classification of granular time series*, *Soft Computing* 19 (4) (2015) 1003–  
797 1017.
- 798 [26] M. Szczuka, A. Jankowski, A. Skowron, D. Slezak, *Building granular*  
799 *systems - from concepts to applications*, in: Y. Yao, Q. Hu, H. Yu,  
800 J. W. Grzymala-Busse (Eds.), *Rough Sets, Fuzzy Sets, Data Mining, and*  
801 *Granular Computing*, Vol. 9437 of *Lecture Notes in Computer Science*,  
802 Springer International Publishing, 2015, pp. 245–255.
- 803 [27] A. Bargiela, W. Pedrycz, *Granular computing: an introduction*, Vol.  
804 717, Springer Science & Business Media, 2012.
- 805 [28] L. Polkowski, P. Artiemjew, *Granular Computing in Decision Approx-*  
806 *imation - An Application of Rough Mereology*, Vol. 77 of *Intelligent*  
807 *Systems Reference Library*, Springer, 2015.
- 808 [29] W. Pedrycz, *Granular computing: Analysis and design of intelligent*  
809 *systems*, CRC Press, 2013.
- 810 [30] G. Nápoles, I. Grau, E. Papageorgiou, R. Bello, K. Vanhoof, *Rough*  
811 *cognitive networks*, *Knowledge-Based Systems* 91 (2016) 46–61.
- 812 [31] B. Kosko, *Fuzzy cognitive maps*, *International Journal of Man-Machine*  
813 *Studies* 24 (1) (1986) 65–75.

- 814 [32] Z. Pawlak, Rough sets, *International Journal of Computer & Informa-*  
815 *tion Sciences* 11 (5) (1982) 341–356.
- 816 [33] G. Nápoles, I. Grau, K. Vanhoof, R. Bello, Hybrid model based on rough  
817 sets theory and fuzzy cognitive maps for decision-making, in: *Rough Sets*  
818 *and Intelligent Systems Paradigms*, Springer, 2014, pp. 169–178.
- 819 [34] G. Nápoles, I. Grau, R. Falcon, R. Bello, K. Vanhoof, A granular intrusion  
820 detection system using rough cognitive networks, in: R. Abielmona,  
821 R. Falcon, N. Zincir-Heywood, H. Abbass (Eds.), *Recent Advances in*  
822 *Computational Intelligence in Defense and Security*, Springer Verlag,  
823 2015, Ch. 7.
- 824 [35] R. Bello, R. Falcon, W. Pedrycz, J. Kacprzyk, *Granular Computing:*  
825 *at the Junction of Rough Sets and Fuzzy Sets*, Springer Verlag, Berlin-  
826 Heidelberg, Germany, 2008.
- 827 [36] A. Abraham, R. Falcon, R. Bello, *Rough Set Theory: a True Landmark*  
828 *in Data Analysis*, Springer Verlag, Berlin-Heidelberg, Germany, 2009.
- 829 [37] Y. Yao, Three-way decision: an interpretation of rules in rough set the-  
830 ory, in: *Rough Sets and Knowledge Technology*, Springer, 2009, pp.  
831 642–649.
- 832 [38] Y. Yao, The superiority of three-way decisions in probabilistic rough set  
833 models, *Information Sciences* 181 (6) (2011) 1080–1096.
- 834 [39] Y. Yao, Three-way decisions with probabilistic rough sets, *Information*  
835 *Sciences* 180 (3) (2010) 341–353.
- 836 [40] B. Kosko, Hidden patterns in combined and adaptive knowledge net-  
837 works, *International Journal of Approximate Reasoning* 2 (4) (1988)  
838 377–393.
- 839 [41] S. Bueno, J. L. Salmeron, Benchmarking main activation functions in  
840 fuzzy cognitive maps, *Expert Systems with Applications* 36 (3) (2009)  
841 5221–5229.
- 842 [42] W. Pedrycz, W. Homenda, From fuzzy cognitive maps to granular cogni-  
843 tive maps, *Fuzzy Systems, IEEE Transactions on* 22 (4) (2014) 859–869.

- 844 [43] W. Pedrycz, The design of cognitive maps: A study in synergy of gran-  
845 ular computing and evolutionary optimization, *Expert Systems with*  
846 *Applications* 37 (10) (2010) 7288–7294.
- 847 [44] W. Pedrycz, A. Jastrzebska, W. Homenda, Design of fuzzy cognitive  
848 maps for modeling time series, *Fuzzy Systems, IEEE Transactions on*  
849 *PP* (99) (2015) 1–1.
- 850 [45] J. C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering  
851 algorithm, *Computers & Geosciences* 10 (2) (1984) 191–203.
- 852 [46] W. Homenda, A. Jastrzebska, W. Pedrycz, Granular cognitive maps re-  
853 construction, in: *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International*  
854 *Conference on, IEEE, 2014*, pp. 2572–2579.
- 855 [47] Y. Zhai, Y.-S. Ong, I. W. Tsang, The emerging “Big Dimensionality”,  
856 *IEEE Computational Intelligence Magazine* 9 (3) (2014) 14–26.
- 857 [48] R. Bryll, R. Gutierrez-Osuna, F. Quek, Attribute bagging: improving  
858 accuracy of classifier ensembles by using random feature subsets, *Pattern*  
859 *recognition* 36 (6) (2003) 1291–1302.
- 860 [49] K. Tumer, J. Ghosh, Classifier combining: analytical results and im-  
861 plications, in: *Working notes from the Workshop Integrating Multi-*  
862 *ple Learned Models, 13th National Conference on Artificial Intelligence,*  
863 *Citeseer, 1996*.
- 864 [50] K. Turner, N. C. Oza, Decimated input ensembles for improved gen-  
865 eralization, in: *Neural Networks, 1999. IJCNN’99. International Joint*  
866 *Conference on, Vol. 5, IEEE, 1999*, pp. 3069–3074.
- 867 [51] L. Breiman, Bagging predictors, *Machine learning* 24 (2) (1996) 123–  
868 140.
- 869 [52] I. H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools*  
870 *and Techniques*, Elsevier, 2011.
- 871 [53] D. R. Wilson, T. R. Martinez, Improved heterogeneous distance func-  
872 tions, *Journal of Artificial Intelligence Research* 6 (1997) 1–34.
- 873 [54] J. Quinlan, Induction of decision trees, *Machine learning* 1 (1986) 81–  
874 106.

- 875 [55] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez,  
876 F. Herrera, Keel data-mining software tool: Data set repository, inte-  
877 gration of algorithms and experimental analysis framework, *Journal of*  
878 *Multiple-Valued Logic and Soft Computing* 17 (255-287) (2010) 11.
- 879 [56] M. Lichman, UCI machine learning repository (2013).  
880 URL <http://archive.ics.uci.edu/ml>
- 881 [57] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Wit-  
882 ten, The weka data mining software: an update, *ACM SIGKDD explo-*  
883 *rations newsletter* 11 (1) (2009) 10–18.
- 884 [58] I. Triguero, S. García, F. Herrera, Self-labeled techniques for semi-  
885 supervised learning: taxonomy, software and empirical study, *Knowl-*  
886 *edge and Information Systems* 42 (2) (2015) 245–284.
- 887 [59] R. Kohavi, The power of decision tables, in: *Machine Learning: ECML-*  
888 *95*, Springer, 1995, pp. 174–189.
- 889 [60] G. H. John, P. Langley, Estimating continuous distributions in bayesian  
890 classifiers, in: *Proceedings of the Eleventh conference on Uncertainty*  
891 *in artificial intelligence*, Morgan Kaufmann Publishers Inc., 1995, pp.  
892 338–345.
- 893 [61] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy, Im-  
894 improvements to platt’s smo algorithm for svm classifier design, *Neural*  
895 *Computation* 13 (3) (2001) 637–649.
- 896 [62] R. Hecht-Nielsen, Theory of the backpropagation neural network, in:  
897 *Neural Networks, 1989. IJCNN.*, International Joint Conference on,  
898 IEEE, 1989, pp. 593–605.
- 899 [63] M. Sumner, E. Frank, M. Hall, Speeding up logistic model tree induc-  
900 tion, in: *Knowledge Discovery in Databases: PKDD 2005*, Springer,  
901 2005, pp. 675–683.
- 902 [64] J. R. Quinlan, *C4.5: programs for machine learning*, Morgan Kauffman  
903 Publishers, 1993.
- 904 [65] J. Su, H. Zhang, A fast decision tree learning algorithm, in: *Proceedings*  
905 *of the 21st National Conference on Artificial Intelligence - Volume 1*,  
906 *AAAI’06*, AAAI Press, 2006, pp. 500–505.

- 907 [66] H. Shi, Best-first decision tree learning, Ph.D. thesis, Citeseer (2007).
- 908 [67] N. Landwehr, M. Hall, E. Frank, Logistic model trees, *Machine Learning*  
909 59 (1-2) (2005) 161–205.
- 910 [68] Y. Amit, D. Geman, Shape quantization and recognition with random-  
911 ized trees, *Neural computation* 9 (7) (1997) 1545–1588.
- 912 [69] D. W. Aha, D. Kibler, M. K. Albert, Instance-based learning algorithms,  
913 *Machine learning* 6 (1) (1991) 37–66.
- 914 [70] J. G. Cleary, L. E. Trigg, et al., K\*: An instance-based learner using  
915 an entropic distance measure, in: *Proceedings of the 12th International*  
916 *Conference on Machine learning*, Vol. 5, 1995, pp. 108–114.
- 917 [71] C. G. Atkeson, A. W. Moore, S. Schaal, Locally weighted learning for  
918 control, in: *Lazy learning*, Springer, 1997, pp. 75–113.
- 919 [72] N. C. Smeeton, Early history of the kappa statistic, *Biometrics* 41 (1985)  
920 795.
- 921 [73] M. Friedman, The use of ranks to avoid the assumption of normality  
922 implicit in the analysis of variance, *Journal of the american statistical*  
923 *association* 32 (200) (1937) 675–701.
- 924 [74] F. Wilcoxon, Individual comparisons by ranking methods, *Biometrics* 1  
925 (1945) 80–93.
- 926 [75] A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests  
927 based on mean-ranks?, *Journal of Machine Learning Research* 17 (2016)  
928 1–10.
- 929 [76] C. Lemke, M. Budka, B. Gabrys, Metalearning: a survey of trends and  
930 technologies, *Artificial Intelligence Review* 44 (1) (2015) 117–130.