

Performance Analysis of Granular versus Traditional Neural Network
Classifiers: Preliminary Results

Peer-reviewed author version

Gerardo, Félix-Benjamín; NAPOLES RUIZ, Gonzalo; Falcon, Rafael; Bello, Rafael & VANHOOF, Koen (2018) Performance Analysis of Granular versus Traditional Neural Network Classifiers: Preliminary Results. In: 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), IEEE,.

DOI: 10.1109/CIVEMSA.2018.8439971

Handle: <http://hdl.handle.net/1942/27925>

Performance Analysis of Granular versus Traditional Neural Network Classifiers: Preliminary Results

Gerardo Felix*, Gonzalo Nápoles[†], Rafael Falcon^{‡§}, Rafael Bello* and Koen Vanhoof[†]

* Department of Computer Science, Central University of Las Villas, Cuba

[†] Faculty of Business Economics, Hasselt Universiteit, Belgium

[‡] Research and Engineering Division, Larus Technologies Corporation, Ottawa, Canada

[§] School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Email: gonzalo.napoles@uhasselt.be

Abstract—A recent trend in Machine Learning is to augment the transparency of traditional classification models using Granular Computing techniques. This approach has been found particularly useful in the neural networks field since most successful neural systems often require complex structures to behave like universal approximators. However, there is a widely-held view stating that, to build an interpretable classifier, one might have to sacrifice some prediction accuracy. We want to challenge this belief by exploring the performance of a recently introduced granular classifier termed *Fuzzy-Rough Cognitive Networks* against low-level (i.e., traditional) neural networks. The simulation results have shown that this neural system can attain quite competitive prediction rates while featuring a shallow, granular architecture. As a bigger picture, this study paves the way for conducting a more thorough evaluation of granular versus low-level neural classifiers in the near future.

Index Terms—Fuzzy-Rough Cognitive Networks, Granular Computing, Neural Systems, Pattern Classification.

I. INTRODUCTION

A plethora of real-world scenarios are concerned with discerning or discriminating among multiple patterns or classes of objects. *Pattern classification* [1] is thus one of the most vibrant areas within modern Artificial/Computational Intelligence. In its canonical formulation, this problem consists of identifying the correct category for each object under consideration from those in a predefined set.

The literature offers a broad array of classification models that tackle this problem from various angles [2] [3], although many of these classifiers do not provide any inherent introspection mechanism to explain how they arrived at a certain conclusion (decision class). A notorious example is that of *Artificial Neural Networks* (ANNs), which are known to be accurate classifiers in general and yet behave as *black boxes* in the sense that neither its neurons nor their connections bear any clear meaning for the problem at hand [4].

A recent trend in the Machine Learning community to boost the interpretability of traditional classification models consists in their augmentation with more symbolic, human-centric constructs, also referred to as *information granules*. *Granular Computing* (GrC) [5] [6] [7] is a research discipline devoted to the construction, representation and processing of information granules. The incorporation of key GrC principles

into the design of these classification methods has given rise to the so called *granular classifiers* [8] [9] [10] [11]. Despite the promising results attained in terms of enhanced interpretability and reduced computational cost, these granular models are rarely compared against their numerical, traditional counterparts on accuracy grounds. Perhaps the underlying belief is that, by representing low-level, numerical information at a higher degree of abstraction, the ensuing reasoning process might not prove competitive accuracy-wise at the numerical (not granular) level.

In this paper we want to challenge this thinking by comparing the classification performance achieved by a recently proposed granular neural classifier, namely *Fuzzy-Rough Cognitive Networks* (FRCNs) [12], against 12 well-established ANN-based classifiers across 123 pattern classification datasets of varying dimensionality, noise levels and incompleteness degrees. This heterogeneity facilitated gaining a global understanding of the performance of these classifiers on the selected datasets by using classification accuracy as the performance metric and avoiding parameter tuning to reduce the computational complexity of this evaluation. All results are supported by a statistical analysis of their significance and they point to the fact that FRCNs are as accurate as the most successful ANN model and certainly capable of beating most of the neural models under consideration while retaining their ability to elucidate their inference process driven by information granules borrowed from Rough Set Theory (RST) [13] [14].

This paper makes the following contributions: (1) we motivate the development of rough cognitive mapping and showcase FRCNs as as promising realization of this concept; (2) encouraged by the superior performance exhibited by FRCNs in [12] with respect to classification methods from multiple families, we hone in on their comparison to ANN-based classifiers in a preliminary attempt to understand how granular neural classifiers behave against their classical peers; (3) we conduct an extensive analysis and discuss the obtained results from a statistically sound perspective and (4) we pave the way for a more thorough evaluation of granular vs. low-level neural classification models in the near future.

The rest of this paper is organized as follows. Section II elaborates on rough cognitive mapping and Section III introduces FRCNs as an embodiment of this concept. The experimental

design and empirical discussion comparing FRCNs to a host of traditional and more advanced neural networks for classification problems are unveiled in Sections IV and V, respectively. Concluding remarks are given in Section VI.

II. ROUGH COGNITIVE MAPPING

In this section, we elaborate on the process of building and exploiting a *Rough Cognitive Network* (RCN) [15], referred to as *rough cognitive mapping*. This process involves three steps, namely: 1) information granulation, 2) network construction, and 3) network initialization and exploitation.

A. Information Granulation

The first step towards building an RCN is to generate a partition $X = \{X_1, \dots, X_k, \dots, X_K\}$ of the universe of discourse \mathcal{U} according to the decision attribute d , such that each subset X_k comprises those objects labeled with the k th decision class. In RST [13] [14], any subset $X_k \subseteq \mathcal{U}$ can be approximated by two crisp sets, which are referred to as its *lower* and *upper approximations* and defined as $\underline{\Phi}X_k = \{x \in \mathcal{U} \mid [x]_{\Phi} \subseteq X_k\}$ and $\overline{\Phi}X_k = \{x \in \mathcal{U} \mid [x]_{\Phi} \cap X_k \neq \emptyset\}$, respectively.

In this classic definition, the equivalence class $[x]_{\Phi}$ comprises the set of objects in \mathcal{U} that are deemed inseparable from x according to the information contained in the attribute subset $\Phi \subseteq \Psi$. In presence of numerical attributes, the equivalence relation may become too strict, so it can be replaced with a weaker binary relation (e.g., a similarity relation).

The lower and upper approximations are used to derive the positive, negative and boundary regions of X_k . The *positive region* $POS(X_k) = \underline{\Phi}X_k$ contains those objects that are surely related to X_k ; the *negative region* $NEG(X_k) = \mathcal{U} - \overline{\Phi}X_k$ denotes those objects that are surely unrelated to X_k , while the *boundary region* $BND(X_k) = \overline{\Phi}X_k \setminus \underline{\Phi}X_k$ contains those objects that might be related to X_k . These information granules (i.e., the three rough approximation regions) convey valuable meaning that can be effectively exploited during the classification of new observations.

B. Designing the Network Topology

Without loss of generality, RCNs can be defined as recurrent neural networks where input neurons represent rough approximation regions, the output neurons are the decision classes and graph edges denote causality relations among these neurons/concepts. Therefore, once the neurons have been determined, we can define the weight matrix according to the following rules:

- (R_1) IF C_i is P_k AND C_j is D_k THEN $w_{ij} > 0$
- (R_2) IF C_i is P_k AND C_j is $D_{v \neq k}$ THEN $w_{ij} < 0$
- (R_3) IF C_i is N_k AND C_j is $P_{v \neq k}$ THEN $w_{ij} < 0$
- (R_4) IF C_i is N_k AND C_j is D_k THEN $w_{ij} < 0$

where C_i and C_j denote two neurons, P_k and N_k are the positive and negative regions corresponding to the k th decision class, respectively, while $w_{ij} \in \mathbb{R}$ is the causal weight between the C_i cause neuron and the C_j effect neuron. We can derive another rule defining the interaction between boundary regions and decision classes since an object $x \in BND(X_k)$ could be

still associated with the k th class. However, the inclusion of this rule rarely leads to higher prediction rates in practice.

C. Network Initialization and Exploitation

To exploit the RCN given a new object x to be classified, we compute the initial activation value for each input neuron according to the function $\mathcal{A} : \mathcal{U} \rightarrow [0, 1]$. This can be done by using the similarity class $\bar{R}(x)$ associated with the target object x as shown in Equation (1):

$$Pr(X|\bar{R}(x)) = \frac{|\bar{R}(x) \cap X_k|}{|X|} \quad (1)$$

which computes the conditional probability of x being a member of the concept X given $\bar{R}(x)$, where $R : xRy \iff \varphi(x, y) \geq \xi$ denotes a similarity relation defined over the distance function $\varphi(x, y)$ and the similarity threshold ξ . Rules R_5 - R_7 formalize the calculation of the activation value of each input neuron according to this principle,

- (R_5) IF C_i is P_k THEN $\mathcal{A}_x^{(0)}(P_k) = \frac{|\bar{R}(x) \cap POS(X_k)|}{|POS(X_k)|}$
- (R_6) IF C_i is N_k THEN $\mathcal{A}_x^{(0)}(N_k) = \frac{|\bar{R}(x) \cap NEG(X_k)|}{|NEG(X_k)|}$
- (R_7) IF C_i is B_k THEN $\mathcal{A}_x^{(0)}(B_k) = \frac{|\bar{R}(x) \cap BND(X_k)|}{|BND(X_k)|}$.

Once the input neurons have been activated, we iteratively propagate the initial conditional probability across the whole cognitive network using Kosko's reasoning rule [16] [17] in Equation (2). The reader may notice that we have intentionally suppressed the $i \neq j$ constraint to prevent neurons from overwriting their own initial value. This relation should not be understood as an authentic causal trigger since an event should not be caused by itself, although it could certainly be influenced by its previous state.

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

The updating process is performed until either the network converges to a fixed-point attractor (equilibrium point) or a maximum number of iterations is reached. At this point, the output neuron (decision class) with the highest activation value is assigned to the observation x .

III. FUZZY-ROUGH COGNITIVE NETWORKS

A pivotal issue when constructing an RCN is related to adjusting the similarity threshold ξ . This hyperparameter determines whether two objects are similar or not, which then influences the construction of the similarity classes upon which the three rough approximation regions are built. Regrettably, small fluctuations of this threshold might lead to quite different conclusions after performing the neural update rule. To deal with this sensitive issue, Nápoles et al. [12] proposed the *fuzzy-rough cognitive networks*, which replaces the crisp information granules with fuzzy-rough ones.

A. Rough Granules with Soft Boundaries

Fuzzy rough sets [18] [19] allow categorizing objects into information granules with soft boundaries, so a strict similarity threshold is no longer required to build a similarity relation. Next, the basis of this theory is presented.

Let $X \in \mathcal{U}$, $(X, \mu_X(x))$ be a fuzzy set and R a fuzzy binary relation, where $\mu_X(x)$ and $\mu_R(y, x)$ are their respective membership functions. The function $\mu_X : \mathcal{U} \rightarrow [0, 1]$ computes the degree to which $x \in \mathcal{U}$ is a member of X , whereas $\mu_R : \mathcal{U} \times \mathcal{U} \rightarrow [0, 1]$ quantifies the degree to which y is presumed to be a member of X from the evidence that x is a member of the fuzzy set X . For the sake of simplicity, $R(x)$ is defined by its membership function, that is, $\mu_{R(x)}(y) = \mu_R(y, x)$.

When defining the *fuzzy lower approximation*, we should consider the degree to which x is a member of X_k under the knowledge in R . This can be measured by the truth value of the statement ‘ $y \in R(x)$ implies $y \in X_k$ ’ under fuzzy sets $R(x)$ and X_k . The truth value can be computed by means of a properly defined necessity measure $\inf_{y \in \mathcal{U}} \mathcal{I}(\mu_R(y, x), \mu_{X_k}(y))$ and implication function $\mathcal{I} : [0, 1] \times [0, 1] \rightarrow [0, 1]$. In this research, we adopt the Lukasiewicz fuzzy operators in all cases. Equation (3) formalizes the membership function defining the k th fuzzy lower approximation.

$$\mu_{R_*(X_k)}(x) = \min \left\{ \mu_{X_k}(x), \inf_{y \in \mathcal{U}} \mathcal{I}(\mu_R(y, x), \mu_{X_k}(y)) \right\} \quad (3)$$

Similarly, we can obtain a membership function for the *fuzzy upper approximation*. To do so, we should gauge the truth value of statement ‘ $\exists y \in \mathcal{U}$ such that $x \in R(y)$ ’ under fuzzy sets $R(x)$ and X_k . The true value of this statement can be computed via a properly defined possibility measure $\sup_{y \in \mathcal{U}} \mathcal{T}(\mu_R(x, y), \mu_{X_k}(y))$ and conjunction function $\mathcal{T} : [0, 1] \times [0, 1] \rightarrow [0, 1]$. Equation (4) formalizes the membership function for the k th fuzzy upper approximation.

$$\mu_{R^*(X_k)}(x) = \max \left\{ \mu_{X_k}(x), \sup_{y \in \mathcal{U}} \mathcal{T}(\mu_R(x, y), \mu_{X_k}(y)) \right\} \quad (4)$$

Based on the fuzzy-rough approximations, we can enunciate the membership functions associated with the fuzzy-rough positive region as $\mu_{POS(X_k)}(x) = \mu_{R_*(X_k)}(x)$, the negative region as $\mu_{NEG(X_k)}(x) = 1 - \mu_{R_*(X_k)}(x)$ and the boundary region as $\mu_{BND(X_k)}(x) = \mu_{R^*(X_k)}(x) - \mu_{R_*(X_k)}(x)$.

Figure 1 displays the FRCN topology for any binary classification problem. Observe that, unlike traditional neural networks whose complexity depends on the number of input attributes, the number of neurons in an FRCN is rather determined by the number of decision classes. This shows the advantages of hybridizing existing neural approaches with granular paradigms towards solving complex problems.

The fuzzy-rough information granules suppress the need for a similarity threshold hyperparameter. This implies that abrupt transitions between classes are replaced with smoother ones, therefore allowing an element to belong to several decision classes with different membership grades.

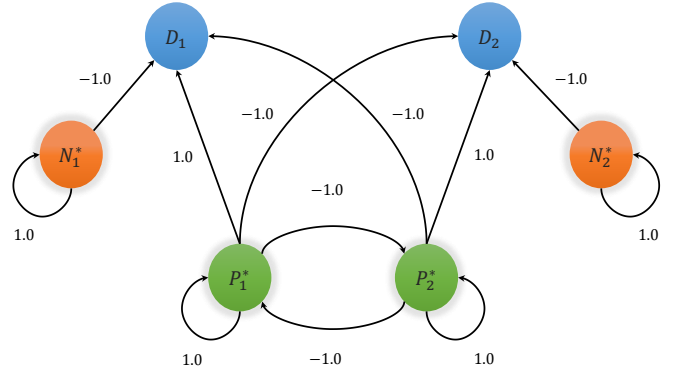


Fig. 1. Fuzzy-Rough Cognitive Network for binary classification problems.

B. Fuzzy Activation Mechanism

Aiming at activating the input neurons in an FRCN, we use the similarity degree between the unlabeled object x and $y \in \mathcal{U}$ as well as the membership degree of y to each fuzzy-rough granular region. In a nutshell, the excitation of the k th input neuron is given by the inclusion degree of the fuzzy intersection set into the k th fuzzy-rough region.

Equations (5) and (6) mathematically formalize the calculation of the activation value of the k th positive and negative neuron, respectively, where \mathcal{T}_2 is a t-norm and $\varphi(y, x)$ is the similarity degree between y and x . It should be mentioned that the t-norm \mathcal{T}_2 could be different from the t-norm adopted to generate the fuzzy-rough granules.

$$A_x(P_k^*) = \frac{\sum_{y \in \mathcal{U}} \mathcal{T}_2(\varphi(y, x), \mu_{POS(X_k)}(y))}{\sum_{y \in \mathcal{U}} \mu_{POS(X_k)}(y)} \quad (5)$$

$$A_x(N_k^*) = \frac{\sum_{y \in \mathcal{U}} \mathcal{T}_2(\varphi(y, x), \mu_{NEG(X_k)}(y))}{\sum_{y \in \mathcal{U}} \mu_{NEG(X_k)}(y)} \quad (6)$$

Once the reasoning process is completed, the label of the output neuron with the highest activation value is assigned to the object x . In this scheme, ties are broken by using a nearest neighbor approach: the model chooses the closest instance (i.e., an object from the universe) to the test object and returns its corresponding decision class.

IV. EXPERIMENTAL DESIGN

We conduct several experiments in order to evaluate our granular model (FRCN)’s prediction capabilities against those of 12 well-established neural classifiers.

We did not undertake a parametric tuning stage for each algorithm in each dataset for two main reasons: (i) our goal is to characterize general performance trends in granular vs. non-granular neural classifiers and whether an ‘‘off-the-shelf’’ granular implementation with default parameter values could perform better than or comparable to a non-granular neural net under the same conditions when tested on 123 datasets of diverse characteristics and (ii) to avoid the steep computational cost entailed by parametric tuning.

A. Dataset Characterization

In our simulations, we employed 123 well-known pattern classification datasets¹ taken from the KEEL [20] and UCI Machine Learning (ML) [21] repositories. These ML problems exhibit different characteristics in order to make the empirical evaluation as extensive as possible. As a summary, the number of attributes ranges from 2 to 262, the number of decision classes from 2 to 24, and the number of instances goes from 14 to 5,644. Likewise, there are 13 noisy and 40 imbalanced datasets, where the imbalance ratio ranges from 5:1 to 439:1. Missing values were imputed by calculating either the mean or the mode depending on whether the attribute was numerical or nominal, respectively. Numerical attributes were normalized to the [0,1] interval before applying 10-fold cross-validation. We used Weka 3.6.11's *normalization filter* that modifies a value x as follows: $x = (x - \min) / (\max - \min) * \text{scale} + \text{translation}$ and we set $\text{scale} = 1$ and $\text{translation} = 0$.

B. Benchmarking Algorithms

The neural classifiers selected for benchmarking purposes in this study are:

- 1) *FRCNs* [12]: our granular recurrent neural network model
- 2) *Genetic Algorithm with Neural Network* (GANN) [22]: Uses a traditional Genetic Algorithm to learn a neural network architecture and backpropagation learning in an individual network to evaluate its fitness.
- 3) *Neural Network Evolutionary Programming* (NNEP) [23]: It relies on product-unit neural networks. Product-units are based on multiplicative nodes instead of additive ones. This model employs Evolutionary Programming to determine the basic structure of the product-unit model and estimate the model coefficients. It uses a softmax transformation as the decision rule and a cross-entropy error function.
- 4) *Radial Basis Function Neural Network* (RBFN-k) [24]: Employs multivariable functional interpolation using radial basis functions.
- 5) *Decremental RBF Neural Network* [24] (Decr-RBFN)
- 6) *Evolutionary RBF Neural Networks* [25] (EvRBFN): used to automatically build an RBFNN.
- 7) *Learning Vector Quantization* (LVQ) [26]: based on the nearest prototype classification rule.
- 8) *Multilayer Perceptron with Backpropagation training* [27] (MLP-BP): one of the classical feedforward neural net models.
- 9) *Multilayer Perceptron with backpropagation training and sigmoid hidden neurons* [28] (MLP-w)
- 10) *Multilayer Perceptron with conjugate gradient-based training* [29] (MLP-CG): a feedforward model based on a conjugate gradient optimization approach.
- 11) *Improved Resilient Backpropagation Plus* [30] (iRProp+): Improved Rprop, an optimization method and one of the

best-performing, first-order learning methods for neural networks.

- 12) *Boosting Ensemble* [31]: A classifier ensemble based on nonlinear projections and boosting instead of using a random subspace, i.e., it takes into account the instances which have posed most difficulties to previous classifiers.
- 13) *RBF Network* (RBFNw) [32] with normalized Gaussian radial basis function network (RBFN-w).

We used the implementations in Weka 3.6.11 [3] and KEEL 3.0 [20]. These two frameworks are free and extensible to new algorithms, hence their increasing popularity.

As already mentioned, the default parameter values for each model have been retained throughout the simulations, i.e., no hyperparameter tuning was carried out. As Triguero et. al. [33] stated, a good choice of parameters increases the algorithms' performance over different datasets; however, a robust classifier should produce good-enough results even when its parameters might not have been optimized for a specific dataset. Moreover, the goal behind our numerical simulations is to examine the prediction capability of these neural net models, even when no algorithm undergoes parametric tuning.

It is worth mentioning that, in the case of the FRCN method, we employed the *Heterogeneous Value Difference Metric* [34] as the standard dissimilarity function.

V. EMPIRICAL DISCUSSION

To statistically analyze the empirical results obtained in our study, we adopted some of the nonparametric tests described in [35] to compare the performance of two or more algorithms on a number of datasets.

The first experiment aimed at determining the existence of statistically significant performance differences among all the classifiers through the Friedman two-way analysis of variances by ranks [36]. The Friedman test is a multiple-comparison nonparametric statistical method that detects whether at least two of the samples in a group represent populations with different median values or not and it also builds a ranking of the algorithms in which the higher the number, the worse its performance. For this test, we used the correction put forth by Iman and Davenport [37] which is more powerful than the Friedman test as well as their corresponding post hoc tests [35]. Figure 2 portrays the average accuracy across all datasets for each neural classifier considered in this study.

For this experiment, the multiple-comparison test suggests rejecting the null hypothesis with a Friedman p-value of 1.97E-10 and Iman and Davenport's p-value of 2.01E-178. Both p-values are smaller than the 5% significance level, hence advocating for the existence of statistically significant performance differences between at least two algorithms across all the selected datasets. Figure 3 displays the rank values computed by the Friedman test. Notice that MLPw is the best-ranked algorithm, FRCN came up as the runner up while MLP-BP is the worst one.

The next step is to determine which of the pairwise comparisons in the group yield statistically significant differences. Figures 4 and 5 unveil the results of three post-hoc procedures

¹The characterization of these datasets can be found at <https://github.com/rfalcon/CIVEMSA-2018-materials/blob/master/FRCN-datasets.pdf>

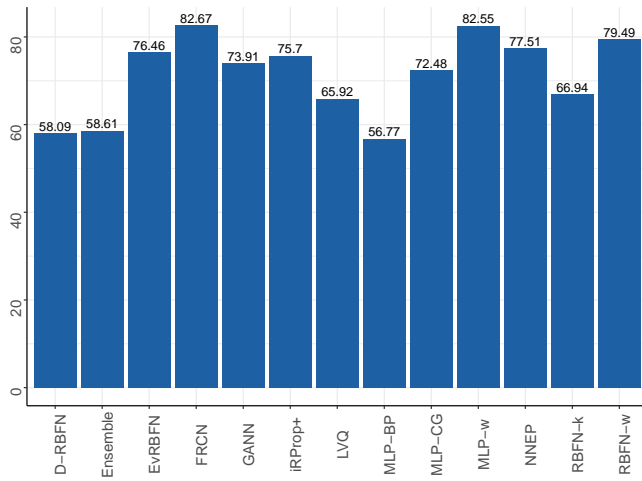


Fig. 2. Average accuracy (%) achieved by each neural network classifier

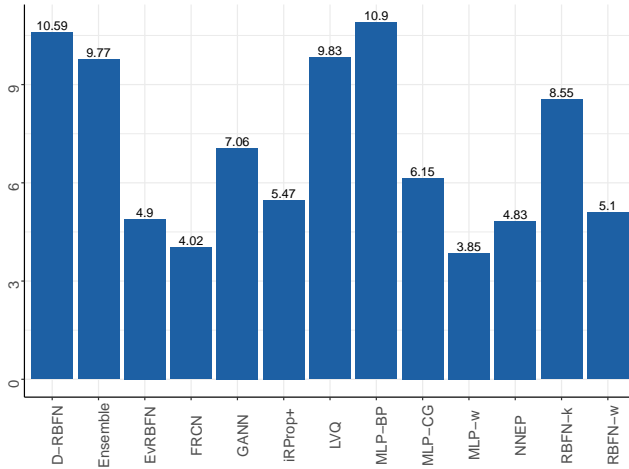


Fig. 3. Friedman rank for each neural network classifier

(Shaffer and Holm, respectively) and report the Friedman signed rank test. In these figures, a horizontal thick line is drawn on all those algorithms that do not have significant performance differences among them (based on the matrix of adjusted p -values by each test at a 5% significance level).

We assume that the null hypothesis can be rejected if at least one of the adopted post-hoc procedures supports the rejection. The results indicate that MLP-w is the best-performing classifier in our study and that no significant performance differences were spotted between FRCN, RBFNw, EvRBFN and NNEP, as the null hypothesis was accepted in each of these pairwise comparisons. Notice, however, that FRCN reports a slightly higher accuracy but is worse ranked than MLP-w and even so produces a similar behavior, which confirms that FRCN is a very competitive neural classifier.

In other words, these experiments reveal that FRCNs are capable of performing like the state-of-the-art, black box classifier MLP-w adopted in the simulations. It is worth emphasizing

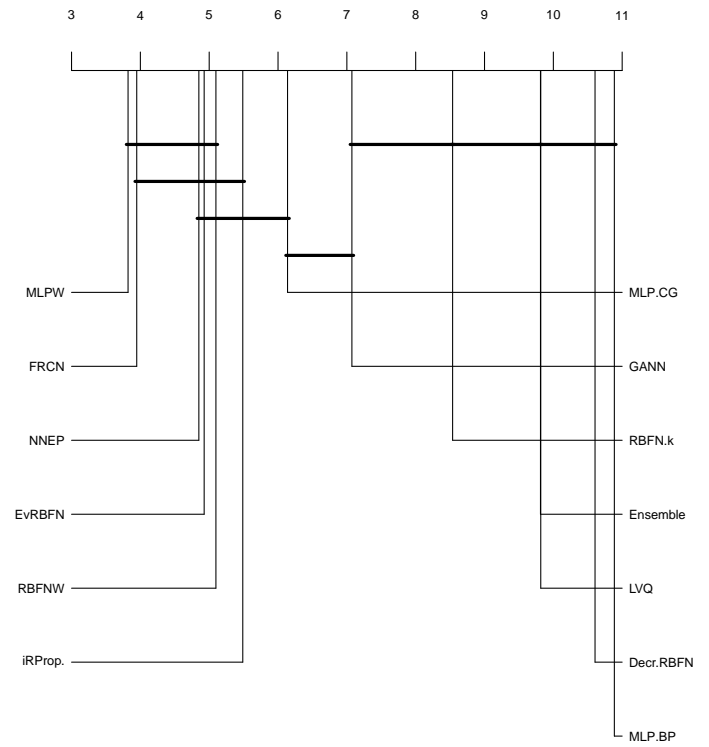


Fig. 4. Statistically significant differences according to Shaffer post-hoc test

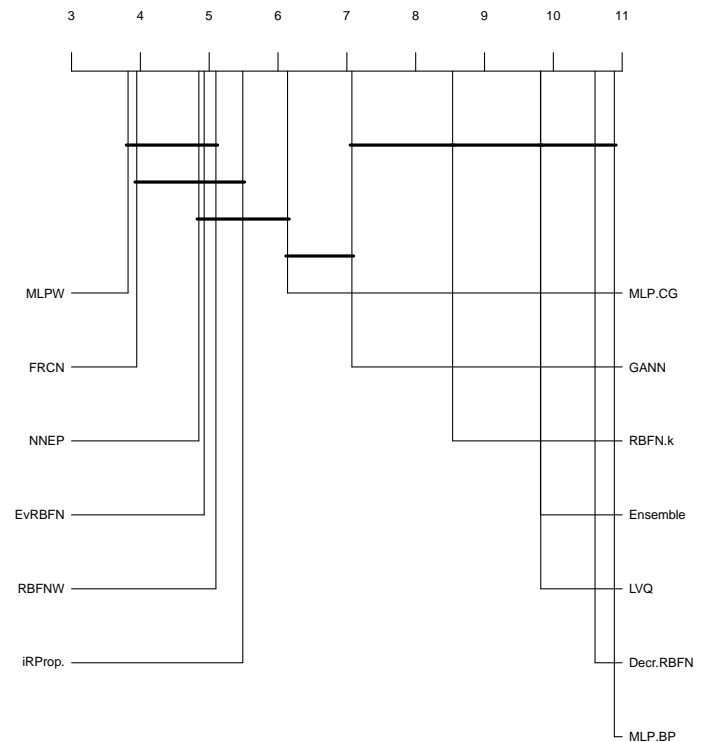


Fig. 5. Statistically significant differences according to Holm post-hoc test

that these implementations have a high computational cost and the main advantage of the FRCN with respect to the ANN-

based traditional models lies in its transparency. In general terms, the FRCN classifier deserves the attention of pattern recognition researchers owing to its competitive results and its transparent inference process.

VI. CONCLUDING REMARKS

In this paper, we have studied the prediction power of the FRCN algorithm against traditional (low-level) neural network classifiers. The FRCN system uses information granules in the form of rough sets with soft boundaries, where an exhaustive parameterization seems to be not required. A pivotal feature that motivated the authors to select this model in this preliminary analysis is that the topology attached to the FRCN network does not scale up with the number of attributes, but with the number of decision classes. On the other hand, the fact that the weight matrix can be defined beforehand (based on the semantics of fuzzy-rough granules) shows the advantages of having meaningful information constructs.

The numerical results have shown that the FRCN algorithm is a strong neural candidate when solving pattern classification problems. It is particularly interesting the ability of this model to perform comparably with MLP-w (which is well-known to be an accurate classifier) while clearly outperforming the remaining methods adopted for comparison. Notice that the weights in FRCNs are not optimized in any way, so it would be reasonable to expect even higher prediction rates if we equip this model with a supervised tuning procedure. Equally, FRCNs could benefit from a learning algorithm capable of adjusting the distance function to existing data. As a future work, we intend to expand our experimental analysis with more diverse granular models in order to empirically verify that granular neural networks with simpler topologies do not necessarily lead to systems with lower prediction rates.

REFERENCES

- [1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.
- [2] C. C. Aggarwal, *Data classification: algorithms and applications*. CRC Press, 2014.
- [3] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
- [4] J. D. Olden and D. A. Jackson, "Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks," *Ecological modelling*, vol. 154, no. 1, pp. 135–150, 2002.
- [5] W. Pedrycz, A. Skowron, and V. Kreinovich, *Handbook of granular computing*. John Wiley & Sons, 2008.
- [6] R. Bello, R. Falcon, W. Pedrycz, and J. Kacprzyk, *Granular computing: at the junction of rough sets and fuzzy sets*. Berlin-Heidelberg, Germany: Springer Verlag, 2008.
- [7] A. Bargiela and W. Pedrycz, *Granular computing: an introduction*. Springer Science & Business Media, 2012, vol. 717.
- [8] R. Al-Hmouz, W. Pedrycz, A. Balamash, and A. Morfeq, "From data to granular data and granular classifiers," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 432–438.
- [9] A. Balamash, W. Pedrycz, R. Al-Hmouz, and A. Morfeq, "Granular classifiers and their design through refinement of information granules," *Soft Computing*, vol. 21, no. 10, pp. 2745–2759, 2017.
- [10] F. M. Bianchi, S. Scardapane, A. Rizzi, A. Uncini, and A. Sadeghian, "Granular computing techniques for classification and semantic characterization of structured data," *Cognitive Computation*, vol. 8, no. 3, pp. 442–461, 2016.
- [11] X. Hu, W. Pedrycz, and X. Wang, "Granular fuzzy rule-based models: A study in a comprehensive evaluation and construction of fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1342–1355, 2017.
- [12] G. Nápoles, C. Mosquera, R. Falcon, I. Grau, R. Bello, and K. Vanhoof, "Fuzzy-rough cognitive networks," *Neural Networks*, vol. 97, pp. 19–27, 2018.
- [13] Z. Pawlak, "Rough sets," *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.
- [14] A. Abraham, R. Falcon, and R. Bello, *Rough set theory: a true landmark in data analysis*. Springer Verlag, 2009, vol. 174.
- [15] G. Nápoles, I. Grau, E. Papageorgiou, R. Bello, and K. Vanhoof, "Rough cognitive networks," *Knowledge-Based Systems*, vol. 91, pp. 46–61, 2016.
- [16] B. Kosko, "Hidden patterns in combined and adaptive knowledge networks," *International Journal of Approximate Reasoning*, vol. 2, no. 4, pp. 377–393, 1988.
- [17] G. Felix, G. Nápoles, R. Falcon, W. Froelich, K. Vanhoof, and R. Bello, "A review on methods and software for fuzzy cognitive maps," *Artificial Intelligence Review*, Aug 2017. [Online]. Available: <https://doi.org/10.1007/s10462-017-9575-1>
- [18] D. Dubois and H. Prade, "Rough fuzzy sets and fuzzy rough sets," *International Journal of General Systems*, vol. 17, pp. 91–209, 1990.
- [19] M. Inuiguchi, W.-Z. Wu, C. Cornelis, and N. Verbiest, *Fuzzy-Rough Hybridization*. Springer Berlin Heidelberg, 2015, pp. 425–451.
- [20] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 255–287, p. 11, 2010.
- [21] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [22] G. F. Miller, P. M. Todd, and S. U. Hegde, "Designing neural networks using genetic algorithms," in *ICGA*, vol. 89, 1989, pp. 379–384.
- [23] F. J. Martínez-Estudillo, C. Hervás-Martínez, P. A. Gutiérrez, and A. C. Martínez-Estudillo, "Evolutionary product-unit neural networks classifiers," *Neurocomputing*, vol. 72, no. 1, pp. 548–561, 2008.
- [24] D. Lowe, "Multi-variable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355.
- [25] V. M. Rivas, J. Merelo, P. Castillo, M. G. Arenas, and J. Castellano, "Evolving rbf neural networks for time-series forecasting with evrbf," *Information Sciences*, vol. 165, no. 3, pp. 207–220, 2004.
- [26] J. C. Bezdek and L. I. Kuncheva, "Nearest prototype classifier designs: An experimental study," *International journal of Intelligent systems*, vol. 16, no. 12, pp. 1445–1473, 2001.
- [27] R. Rojas, *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.
- [28] R. Hecht-Nielsen *et al.*, "Theory of the backpropagation neural network," *Neural Networks*, vol. 1, no. Supplement-1, pp. 445–448, 1988.
- [29] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [30] C. Igel and M. Hüsken, "Empirical evaluation of the improved rprop learning algorithms," *Neurocomputing*, vol. 50, pp. 105–123, 2003.
- [31] N. García-Pedrajas, C. García-Osorio, and C. Fyfe, "Nonlinear boosting projections for ensemble construction," *Journal of Machine Learning Research*, vol. 8, no. Jan, pp. 1–33, 2007.
- [32] G. Bugmann, "Normalized gaussian radial basis function networks," *Neurocomputing*, vol. 20, no. 1-3, pp. 97–110, 1998.
- [33] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 245–284, 2015.
- [34] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *Journal of Artificial Intelligence Research*, 1997.
- [35] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.
- [36] D. J. Sheskin, *Handbook of parametric and nonparametric statistical procedures*. crc Press, 2003.
- [37] R. L. Iman and J. M. Davenport, "Approximations of the critical region of the Friedman statistic," *Communications in Statistics-Theory and Methods*, vol. 9, no. 6, pp. 571–595, 1980.