UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

Masterthesis

Increasing the Performance of Fuzzy-Rough Cognitive Networks

Marnick Vanloffelt

Scriptie ingediend tot het behalen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

PROMOTOR :

dr. Gonzalo NAPOLES RUIZ

2018
2019

UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Faculteit Bedrijfseconomische Wetenschappen

master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

Masterthesis

Increasing the Performance of Fuzzy-Rough Cognitive Networks

Marnick Vanloffelt

Scriptie ingediend tot het behalen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

PROMOTOR :

dr. Gonzalo NAPOLES RUIZ

Fuzzy-Rough Cognitive Networks: Building Blocks and their Contribution to Performance

Marnick Vanloffelt[1], Gonzalo Nápoles[1]

Abstract

Pattern classification is a popular research field within the Machine Learning discipline. Black-box models have proven to be potent classifiers in this particular field. However, their inability to provide a transparent decision mechanism is often regarded as an undesirable feature. *Fuzzy-Rough Cognitive Networks* are granular classifiers that have proven competitive and effective in such tasks. In this paper, we examine the contribution of the FRCN's main building blocks, being the causal weight matrix and the activation values of the neurons, to the model's average performance. Noise injection is employed to this end. Furthermore, we explore various alternatives for the current structure of these building blocks. Firstly, we experiment with possible ways of adjusting the weight matrix, which is originally composed of fixed weight values based on set rules. Secondly, we explore if computing a confidence degree per decision class from another, potentially weaker, classifier could lead to more powerful neuron activation and possibly an improved performance.

Keywords: pattern classification, fuzzy-rough sets, granular classifiers, fuzzy rough cognitive networks

1. Introduction

When it comes to *Pattern classification* [1], a wide variety of models and techniques exists to approach the classification problem. However, most of the existing models are not transparent in the sense that they do not explain how they arrived at their conclusions. Therefore, these models are labelled as *black boxes*. *Fuzzy Cognitive Maps* (FCMs), first introduced by Bart Kosko [2], were intended to circumvent the tradeoff between knowledge acquisition and knowledge processing. Kosko used a fuzzy di-graph to represent causal reasoning in a visual and easily interpretable fashion. As such, FCMs are interpretable recurrent neural networks [3] with signed and weighted causal relations between

Email addresses: marnick.vanloffelt@student.uhasselt.be (Marnick Vanloffelt), gonzalo.napoles@uhasselt.be (Gonzalo Nápoles)

[1]Hasselt University

the model's concepts. In turn, these concepts are low-level representations of the underlying data. As such, both the neurons and connections get specific meanings with respect to the classification problem at hand.

Although the transparency displayed in FCMs has proven to be a valuable and desirable feature, FCM-based models have several disadvantages. One of these disadvantages is the initialisation of the weight matrix by experts [4, 5, 6]. Another is the discrete performance of FCM-based classifiers when compared with traditional black boxes. Therefore, Nápoles et al. [3] introduced *Rough Cognitive Networks* (RCNs) in order to create a transparent, yet accurate classifier. These RCNs are granular neural networks, which augment the reasoning scheme of FCMs with information granules derived from *Rough Set Theory* (RST). RST entails the construction of three regions (*positive*, *negative* and *boundary* regions) based on the approximate *similarity* of objects in the universe of discourse [7]. Rough sets essentially establish that an object can belong to different sets or relations at the same time, albeit to varying degrees [8].

Although the RCN algorithm has proven competitive in solving a wide variety of classification problems, the model was still quite sensitive to an input parameter denoting the similarity threshold upon which the rough information granules are built [9]. Therefore, Nápoles et al. [9] introduced the *Rough Cognitive Ensembles* (RCEs), which use a collection of RCNs, each operating at a different granularity level. As such, each RCN employs a different similarity threshold. This ensemble architecture promotes model diversification. Unfortunately, the model's transparency is severely damaged due to this ensemble strategy, similar to Random Forests, for example.

As an alternative to overcoming the parametric learning requirements related to the similarity threshold without damaging the model's transparency and discriminatory power, Nápoles et al. [10] introduced a classifier based on *Fuzzy-Rough Set Theory* (FRST), which is a hybridisation of the Fuzzy Set and Rough Set Theories [11, 12, 13]. In FRST, the rough sets described in Rough Set Theory are extended with fuzzy sets in order to characterise the degree to which an object belongs to an information granule [8]. The new model was consequently named *Fuzzy-Rough Cognitive Network* (FRCN).

FRCNs proved to be superior to RCNs with respect to the model's performance [8]. Furthermore, FRCNs also outperformed various popular traditional classifiers such as Support Vector Machines, Simple Logistic and k-Nearest Neighbours [8]. The model also performed similarly to certain black-box models, such as the Multi-Layer Perceptron, Random Forest and Logistic Model Tree, while simultaneously providing a more translucent decision mechanism due to the transparent high-level model topology [8].

Although FRCNs are promising in terms of performance and transparency, it was noted that a learning algorithm able to compute the weight matrix from data in the underlying FCM is absent [10, 14]. Indeed, the initial weight matrix consists of fixed weights, with a value of either -1.0 or 1.0. The question if such a learning algorithm would provide merit to the FRCNs performance without harming transparency, and if so, which algorithms would be suitable in that regard, remains open. Therefore, in this paper, we investigate whether changing the values of the causal weights affects performance, and if so, in which way.

It is worth noting that the weight matrix is not the only crucial component that could benefit from optimisation. The activation values of input-type neurons are possible candidates as well, as they are a vital building block. In this paper, we explore the effects of adding a confidence measure per decision class, derived form another classifier, to the initial activation values of the input neurons. We will elaborate on the two alternate approaches we employed in this paper to implement this confidence degree later on.

The remainder of this paper is structured as follows. Firstly, we give an overview on the FRCN model in Section 2. Section 3 describes the research questions we intend to tackle, as well as our applied methodology. Next, Section 4 outlines the results obtained from numerical simulations and a discussion based on our analysis. Finally, Section 5 summarises our findings and includes recommendations for further research.

2. Fuzzy-Rough Cognitive Networks: Theoretical Background

The process of constructing and exploiting an FRCN follows three distinct phases: the information space granulation, network construction and network exploitation. In this section, each of these phases is described through their respective theoretical foundations.

2.1. Information Space Granulation

Let us begin with the information space granulation, which essentially entails dividing the available information (in this case a dataset, which we will label as the universe of discourse $\mathcal{U}$) into *granules*. Consider a universe of discourse $\mathcal{U}$, a fuzzy set $X \in \mathcal{U}$ and a fuzzy binary relation $P \in \mathcal{F}(\mathcal{U} \times \mathcal{U})$. Let $\mu_X(x)$ and $\mu_P(y,x)$ be their respective membership functions. The function $\mu_X : \mathcal{U} \to [0,1]$ computes the membership degree to which $x \in \mathcal{U}$ is a member of X, while $\mu_P : \mathcal{U} \times \mathcal{U} \to [0,1]$ denotes the degree to which y is presumed to be a member of X depending on whether x is a member of the fuzzy set X [10]. In this paper, $P(x)$ is defined by its membership function $\mu_{P(x)}(y) = \mu_P(y,x)$ [8].

To define the lower and upper approximations, we employ the measures presented in the possibility theory [15]. Concretely, the truth values of the statements "$y \in P(x)$ implies $y \in X$" and "$\exists y \in \mathcal{U}$ such that $x \in P(y)$" under fuzzy sets $P(x)$ and X, are used to define the lower and upper approximations of a set in fuzzy environments [8]. Let us consider these approximations and their membership functions separately.

Firstly, to construe the membership function of the lower approximation, we use the necessity measure $\inf_{y\in\mathcal{U}} \mathcal{I}(\mu_P(y,x), \mu_{X_k}(y))$ with $\mathcal{I}$ being an implication function such that $\mathcal{I} : [0,1] \times [0,1] \to [0,1]$. This function is used to assess the truth value of the statement "$y \in P(x)$ implies $y \in X$" [8, 10]. Equation 1 formalises this idea as follows:

$$\mu_{P_*(X_k)}(x) = \min \left\{ \mu_{X_k}(x), \inf_{y\in\mathcal{U}} \mathcal{I}(\mu_P(y,x), \mu_{X_k}(y)) \right\}. \tag{1}$$

where X_k denotes the set containing all objects labelled with the k-th decision class. In a comparable fashion, we can derive the membership function for the upper approximation, which is shown in Equation 2,

$$\mu_{P^*(X_k)}(x) = \max\left\{\mu_{X_k}(x), \sup_{y \in \mathcal{U}} \mathcal{T}_1(\mu_P(x,y), \mu_{X_k}(y))\right\}. \quad (2)$$

such that the truth value of the statement "$\exists y \in \mathcal{U}$ such that $x \in P(y)$" can be measured with the possibility measure $\sup_{y\in\mathcal{U}} \mathcal{T}_1(\mu_P(x,y), \mu_{X_k}(y))$ with a conjunction function, or t-norm[2], $\mathcal{T}_1 : [0,1] \times [0,1] \to [0,1]$ [10].

In both Equation 1 and Equation 2, we do not assume that $\mu_P(x,x) = 1, \forall x \in \mathcal{U}$ [16]. Therefore, we compute the minimum between $\mu_X(x)$ and $\inf_{y\in\mathcal{U}} \mathcal{I}(\mu_P(y,x), \mu_{X_k}(y))$ when computing $\mu_{P_*(X_k)}(x)$, and the maximum between $\mu_X(x)$ and $\sup_{y\in\mathcal{U}} \mathcal{T}(\mu_P(x,y), \mu_{X_k}(y))$ when computing $\mu_{P^*(X_k)}(x)$. This feature allows preserving the inclusiveness of $P_*(X)$ in the fuzzy set X and the inclusiveness of X in $P^*(X)$ [10].

Equations (3) and (4) display the membership functions associated with the fuzzy-rough positive and negative regions, respectively,

$$\mu_{POS(X_k)}(x) = \mu_{P_*(X_k)}(x) \quad (3)$$

$$\mu_{NEG(X_k)}(x) = 1 - \mu_{P_*(X_k)}(x). \quad (4)$$

These memberships functions allow computing more flexible information granules by replacing hard transitions between classes with soft ones. This allows an element to belong to more than one decision class, albeit to varying degrees. As such, a strict similarity threshold is no longer required [10].

Next, let us consider $X = \{X_1, \ldots, X_k, \ldots, X_M\}$ with $X \subset \mathcal{U}$ according to the values of the different decision classes. Consequently, $X_k \subseteq X$ comprises those objects labelled as D_k. Based on this partition, we can define the membership degree of $x \in \mathcal{U}$ to a subset X_k, assuming that all objects labelled as D_k have maximum membership degree to the X_k:

$$\mu_{X_k}(x) = \begin{cases} 1 & , y \in X_k \\ 0 & , y \notin X_k \end{cases}. \quad (5)$$

However, more sophisticated variants can be formalised as well, which would allow an object to have a varying degree of membership to different similarity classes at the same time. We define an alternative formulation of the membership degree of an object to its similarity class later on.

Another component to be defined is the membership function $\mu_P(y,x)$ associated with the fuzzy binary relation. Equation (6) shows the function adopted

[2] A t-norm is a conjunction function $\mathcal{T} : [0,1] \times [0,1] \to [0,1]$ that fulfils three conditions: (i) $\forall a \in [0,1], \mathcal{T}(a,1) = \mathcal{T}(1,a) = a$, (ii) $\forall a,b \in [0,1], \mathcal{T}(a,b) = \mathcal{T}(b,a)$, and (iii) $\forall a,b,c \in [0,1], \mathcal{T}(a,\mathcal{T}(b,c)) = \mathcal{T}(\mathcal{T}(a,b),c)$.

in this paper, which depends on the membership degree of object x to X, and the similarity degree between x and y [10, 8]. The similarity degree $\varphi(x,y)$ denotes the complement of the normalised distance $\delta(x,y)$ between two instances x and y. Possible candidates for the distance function are the Heterogeneous Manhattan Overlap Metric (HMOM), Heterogeneous Euclidean Overlap Metric (HEOM) and Heterogeneous Value Difference Metric (HVDM). However, other alternatives to the three functions mentioned above are also available.

$$\mu_P(y,x) = \mu_{X_k}(x)\varphi(x,y) = \mu_{X_k}(x)(1-\delta(x,y)) \quad (6)$$

To summarise the information granulation porcess, let us assume that the universe of discourse $\mathcal{U}$ contains those objects comprised into the training set and $\Theta : \mathcal{U} \rightarrow \mathcal{D}$ is a function that returns the decision class attached to each training set instance, such that $\mathcal{D} = \{D_1, \ldots, D_K\}$. Algorithm 1 summarises the steps for granulating the information space under the fuzzy settings described above.

Algorithm 1. Information space granulation.

```
FOREACH x ∈ 𝒰 DO
    IF Θ(x) = D_k THEN
        X_k ← X_k ∪ {x}
    END IF
    Compute μ_{X_k}(x) according to Equation 5
END
FOREACH x ∈ 𝒰 DO
    FOREACH subset X_k DO
        Compute μ_{POS(X_k)}(x) according to Equation (3)
        Compute μ_{NEG(X_k)}(x) according to Equation (4)
    END
END
```

2.2. Network Construction

After the information space granulation, the resulting Fuzzy-Rough constructs are used to build a recurrent neural network. Similarly to RCNs, input neurons denote positive or negative fuzzy-rough regions and output neurons denote the decision classes for the problem at hand. However, contrary to the RCN's topology, boundary regions are not included, as previous research pointed out that including the these regions into did not significantly increase the classifier's discriminatory ability [8]. This behaviour is not surprising because in crisp-rough environments the hesitant evidence is more conclusive when compared to the evidence coming from fuzzy-rough granules [10]. As such, we construe the neural network topology using the following rules:

- (R_1^*) IF $C_i = P_k^*$ AND $C_j = D_k$ THEN $w_{ij} = 1.0$
- (R_2^*) IF $C_i = N_k^*$ AND $C_j = D_k$ THEN $w_{ij} = -1.0$

- (R_2^*) IF $C_i = P_k^*$ AND $C_j = D_{v \neq k}$ THEN $w_{ij} = -1.0$
- (R_4^*) IF $C_i = P_k^*$ AND $C_j = P_{v \neq k}$ THEN $w_{ij} = -1.0$

where C_i is the i-th neural concept, D_k represents the k-th decision class, and P_k^* and N_k^* are neurons denoting the positive and negative fuzzy-rough region associated to the k-th decision class.

Figure 1 shows the network topology of FRCNs for binary classification problems. With respect to the topological characteristic, any FRCN consists of $2|\mathcal{D}|$ input neurons, $|\mathcal{D}|$ output neurons and $|\mathcal{D}|(4 + |\mathcal{D}|)$ causal weights. As such, the number of neurons is determined by the number of decision classes, as is the number of causal relations.

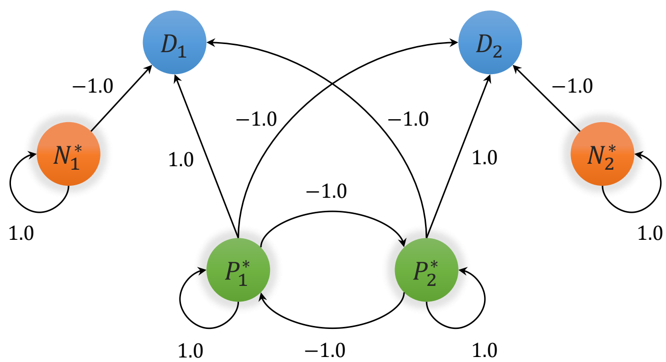


Figure 1: Fuzzy-Rough Cognitive Network for binary classification problems.

Algorithm 2 shows the steps required to build the topology of the granular neural network from the information granules computed in Algorithm 1.

Algorithm 2. Network construction.

```
FOREACH X_k DO
   Add a neuron P_k as the kth positive region
   Add a neuron N_k as the kth positive region
END
FOREACH D_k DO
   Add a neuron D_k as the kth decision
END
FOREACH C_i DO
   FOREACH C_j DO
      Assign w_ij according to rules R_1^* − R_4^*
   END
END
```

2.3. Network Exploitation

Once the network has been constructed, we can classify new (unlabelled) instances by activating the input-type neurons and performing the (high-level) neural reasoning process, which is the same as in Fuzzy Cognitive Maps [10, 4, 5]. In order to activate these neurons, we use the similarity degree between the object y and $x \in \mathcal{U}$ as well as the membership degree of x to each fuzzy-rough granular region. Furthermore, we calculate the inclusion degree of the fuzzy intersection set into the k-th fuzzy-rough region. This procedure produces a normalised value that will be used to activate the input neurons in the causal network [10].

Equation (7) formalises a generalised measure to compute the activation value of the k-th positive neuron, where $\mathcal{T}_2$ denotes a t-norm, $\varphi(x,y)$ is the similarity degree between x and y and $\mu_{POS(X_k)}(x)$ is the membership degree of x to the k-th positive region. Similarly, we can activate neurons denoting fuzzy-rough negative regions. Only output neurons remain inactive at the beginning of the neural reasoning process, as their values depend on the previous activation values of the input-type neurons.

$$\mathcal{A}(P_k^*) = \frac{\int \mathcal{T}_2(\varphi(x,y), \mu_{POS(X_k)}(x))dx}{\int \mu_{POS(X_k)}(x)dx} \tag{7}$$

However, due to the fact that the universe of discourse $\mathcal{U}$ can be described finite due to the fact that most datasets are finite, the use of integrals may not be convenient [10]. Rules (R_5^*) and (R_6^*) show a more practical way of activating the positive and negative fuzzy-rough regions, respectively.

- (R_5^*) IF $C_i = P_k^*$ THEN $A_i^{(0)} = \frac{\sum_{x\in\mathcal{U}} \mathcal{T}_2(\varphi(x,y), \mu_{POS(X_k)}(x))}{\sum_{x\in\mathcal{U}} \mu_{POS(X_k)}(x)}$
- (R_6^*) IF $C_i = N_k^*$ THEN $A_i^{(0)} = \frac{\sum_{x\in\mathcal{U}} \mathcal{T}_2(\varphi(x,y), \mu_{NEG(X_k)}(x))}{\sum_{x\in\mathcal{U}} \mu_{NEG(X_k)}(x)}$

Once the initial activation vector $A^{(0)}$ associated with the object y has been computed, we perform the neural reasoning process until (i) a fixed-point attractor is discovered, or (ii) a maximal number of iterations is reached. At that point, the label of the output neuron having the highest activation value is assigned to the object [10, 8].

Algorithm 3a shows the first step towards exploiting the neural network, the activation of input neurons for a new test instance x, while Algorithm 3b summarises how to determine the decision class from output neurons.

Algorithm 3a. Network activation procedure.

```
FOREACH D_k DO
    Calculate A_x^(0)(P_k) according to rule R_5^*
    Calculate A_x^(0)(N_k) according to rule R_6^*
END
```

Algorithm 3b. Network reasoning procedure.

```
FOR $t = 0$ TO $T$ DO
    $converged \leftarrow TRUE$
    FOREACH $C_i$ DO
        Compute $A_i^{(t+1)} = f\left(\sum_{j=1}^{M} w_{ji} A_j^{(t)}\right)$
        IF $A_i^{(t)} \neq A_i^{(t+1)}$ THEN
            $converged \leftarrow FALSE$
        END
    END
    IF $converged$ THEN
        RETURN $argmax_k\{\mathcal{A}_x^{(t+1)}(D_k)\}$
    END
END
IF not $converged$ THEN
    RETURN $argmax_k\{\mathcal{A}_x^{(T)}(D_k)\}$
END
```

It is worth mentioning that the FRCN algorithm can operate in either a lazy or inductive fashion. In a lazy setting, both the fuzzy-rough granules and the network topology can be constructed when the new instance arrives. This is however not efficient since the granules and the topology can be reused to classify new instances. In the inductive approach, the knowledge is stored into the discovered granules and the causal weight matrix, which is determined by rules (R_1^*) - (R_4^*) [9, 8].

In this Section, we provided the construction and exploitation of an FRCN in three separate stages: information space granulation (Algorithm 1), network construction (Algorithm 2) and network exploitation (Algorithms 3a and 3b) by means of a recurrent neural network. In the next Section, we describe the theoretical contributions of this paper.

3. Research contributions

It was already mentioned that optimising the weight matrix by means of a supervised learning algorithm is a possible research track. Currently, the weights are fixed, with a value of either 1.0 or -1.0. However, the prerequisite for the possible implementation of an algorithm able to compute different values from data, is knowing what the contribution of the weight matrix is with respect to the model's performance. Moreover, the contribution of the different weight sets in the weight matrix should also be examined. This way, a targeted optimisation could be implemented instead of a general one, should the contribution of the weight sets be uneven. As such, the first research question is what the contribution of the individual weight sets is in an FRCN.

Next to weight matrix optimisation, we also mentioned optimising the activation of input-type neurons, as a more powerful activation could lead to increased performance results. There are various ways that could possibly lead to this, but the track we explore in this paper is the definition of an alternative to Equation 5. This equation stipulates that the each object in the universe of discourse $\mathcal{U}$ has maximum membership to the subset X_k, which denotes the objects labelled as D_k. However, our variant would allow an object to have varying membership degrees to different similarity classes at the same time. Equation 8 shows a general version of our first proposal,

$$\mu_{X_k}(x) = \begin{cases} \rho^{\Omega}_{X_k}(x) & , y \in X_k \\ 0 & , y \notin X_k \end{cases} \tag{8}$$

where $\rho^{\Omega}_{X_k}(x)$ denotes the confidence degree of x to be a member of X_k according to the Ω classifier. This Ω can be any classifier able to compute such a confidence degree, even potentially weaker ones compared to the FRCN. In this paper, we present two concrete proposals: a soft covering of the information space and a hybrid FRCN.

Let us start with the first proposal, which is to compute the confidence degree $\rho^{\Omega}_{X_k}(x)$ for and object x and every decision class k, from another, potentially weaker classifier Ω, which can subsequently be used to create a soft covering of the information space instead of a crisp one. As such, an object can belong to different similarity classes at the same time, albeit to varying degrees. This way, we can "correct" distortions in the initial activation values of these neurons caused by instances which are assigned to the wrong decision class *before* the FRCN model is built and exploited. In this variant, we are thus injecting additional knowledge during the granulation process.

Let us consider the following example. Let $x \in X$, with $X \in \mathcal{U}$, be an object labelled with the k-th decision class. Let ϑ be an FRCN and Ω be the classifier used to compute $\rho^{\Omega}_{X_k}(x)$, and thus also $\mu_{X_k}(x)$. If ϑ labels x as D_m, but Ω labels this same object as D_k, with $k \neq m$, then this is a distortion which leads to a decreased performance of the FRCN. However, the implementation of Equation 8 instead of Equation 5 forces the FRCN to take the confidence weights produced by Ω into account, thus allowing it to correct this error.

Our second proposal has a similar purpose to the first one, but employs a very different approach. Instead of providing a soft covering for the information space, we insert the confidence weights directly into the initial activation values of the FRCN's input-type neurons. As such, (R^*_5) and (R^*_6) are replaced by

- $(R^{*'}_5)$ IF $C_i = P^*_k$ THEN $A^{(0)}_i = \frac{\sum_{x \in \mathcal{U}} \mathcal{T}_2(\varphi(x,y), \mu_{POS(X_k)}(x))}{\sum_{x \in \mathcal{U}} \mu_{POS(X_k)}(x)} \rho^{\Omega}_{X_k}(x)$
- $(R^{*'}_6)$ IF $C_i = N^*_k$ THEN $A^{(0)}_i = \frac{\sum_{x \in \mathcal{U}} \mathcal{T}_2(\varphi(x,y), \mu_{NEG(X_k)}(x))}{\sum_{x \in \mathcal{U}} \mu_{NEG(X_k)}(x)} (1 - \rho^{\Omega}_{X_k}(x))$

respectively. This hybrid approach has proven to be effective in FCMs [17], with the sole exception that we refrain from employing black box models to produce the confidence degrees. Evidently, using black-box models to this end is detrimental to the model's transparency. Note that this does not mean that black-box models cannot be used. The only explicit requirement we might want to consider for Ω is related to the computational burden. Therefore, we opt for lighter classifiers in this paper.

The two approaches described above could significantly affect performance, should these models provide additional information, containing complementary insights to perform the classification process.

4. Numerical simulations and discussion

In this section, we describe our applied methodology to tackle the research questions described in the previous Section, the data and benchmark used in the numerical simulations and our experimentation results.

4.1. Methodology

To explore whether FRCNs are susceptible to the implementation of a learning algorithm able to optimise the weight matrix, resulting in an increased performance, noise was gradually introduced to the weight matrix. Furthermore, we also injected Gaussian noise into the activation values. As such, we are able to determine the relevance of both the weight matrix and activation values in the network's performance.

The main hypothesis here is that if noise is introduced to the weight matrix, the performance of the FRCN will gradually yet significantly deteriorate. Again, we hypothesise that if noise is gradually introduced to the initial activation values, performance will significantly decrease.

Noise injection is also used to assess the relevance of the initial activation values of input-type neurons. Similar to the previous approach, Gaussian noise was gradually introduced to these initial activation values. Again, we hypothesise that if noise is gradually introduced to the initial activation values, performance will significantly decrease.

Once we assessed the contribution of both the weight matrix and initial activation values, we move on to the two algorithm proposals presented in Section 3.

To determine the impact of the proposed changes on the model's performance, we used different white box classifiers. In this case, we used the One Rule classifier (OneR) [18], the J48 decision tree (J48) [19], Bayesian Networks (BN) [20], Logistic Regression (Log Reg) and k-Nearest Neighbours classifier (kNN) [21].

Our hypothesis for the two algorithm proposals is that they could significantly affect performance, on condition that these models provide additional (external) information, containing complementary insights to perform the classification process.

Our hypothesis is that the two approaches described above will significantly increase performance, should these models provide additional information, containing complementary insights to perform the classification process.

4.2. Data and benchmark evaluation

When Gaussian noise was inserted, the same seed was used to ensure the reproducibility of the results. In terms of a performance benchmark, we employ the average Cohen's kappa instead of the average accuracy rate. For transparency, we employ the Occam's Razor principle; A simpler topology is preferred over complex model architectures. This principle implicitly guards the model's topological transparency and simplicity, thus indirectly preserving one of its most valuable features.

We resorted to the WEKA software package (v3.6.11) [22] to conduct these simulations. Simulation results were processed in R (v3.5.2) [23]. Furthermore, 10-fold cross validation was applied to all iterations in each separate experiment. Furthermore, we use the Wilcoxon Signed-Rank Test with Holm correction [24, 25] to provide statistical support to the claims made in this paper.

With respect to the datasets, Table 1 displays the 55 academic datasets employed in the numerical simulations [26].

Table 1: Datasets used in the experiments.

ID	Dataset	Instances	Attributes	Noisy	Imbalance
1	acute-inflammation	120	8	No	No
2	acute-nephritis	120	6	No	No
3	anneal	898	38	No	85:1
4	anneal.orig	898	38	No	85:1
5	appendicitis	106	7	No	No
6	audiology	226	69	No	No
7	australian	14	2	No	No
8	autos	205	25	No	22:1
9	balance-noise	625	4	Yes	5:1
10	balance-scale	625	4	No	5:1
11	balloons	16	4	No	No
12	banana	5300	2	No	No
13	blood-transfusion	748	4	No	No
14	breast	277	9	No	No
15	breast-cancer-wisc-prog	198	34	No	No
16	bridges-version1	107	12	No	No
17	bridges-version2	107	12	No	No
18	car	1728	6	No	17:1
19	cleveland	297	13	No	12:1
20	colic	368	22	No	No
21	colic.orig	368	27	No	No
22	collins	500	23	No	13:1
23	contact-lenses	24	4	No	No
24	contraceptive	1473	9	No	No
25	credit-a	690	15	No	No
26	crx	653	15	No	No
27	dermatology	358	34	No	5:1
28	echocardiogram	131	11	No	5:1
29	ecoli	336	7	No	71:1
30	ecoli0	220	7	No	No
31	ecoli-0vs1	220	7	No	No
32	ecoli1	336	7	No	No
33	ecoli2	336	7	No	5:1

Continued on next page

Table 1 – continued from previous page

ID	Dataset	Instances	Attributes	Noisy	Imbalance
34	ecoli3	336	7	No	8:1
35	ecoli-5an-nn	336	7	Yes	71:1
36	energy-y1	768	8	No	No
37	energy-y2	768	8	No	No
38	eucalyptus	738	19	No	No
39	flags	194	28	No	15:1
40	glass	214	9	No	8:1
41	glass0	214	9	No	No
42	glass-0123vs456	214	9	No	No
43	glass1	214	9	No	No
44	glass-10an-nn	214	9	Yes	8:1
45	glass2	214	9	No	No
46	glass-20an-nn	214	9	Yes	8:1
47	glass3	214	9	No	6:1
48	glass-5an-nn	214	9	Yes	6:1
49	glass6	214	9	No	6:1
50	haberman	306	3	No	No
51	iris	150	4	No	No
52	iris0	150	4	No	No
53	iris-10an-nn	150	4	Yes	No
54	iris-20an-nn	150	4	Yes	No
55	iris-5an-nn	150	4	Yes	No

Before we move on to the results, we establish a base case model topology. This base case will serve as the benchmark for all comparisons made in the remainder of this paper. The following fuzzy operators and distance function will be used as the default parameters for the base case FRCN model: the Lukasiewicz fuzzy t-norm for both $\mathcal{T}_1$ and $\mathcal{T}_2$, the Lukasiewicz fuzzy implicator and the HMOM distance function.

4.3. Topology

Firstly, the contribution of the causal weight matrix is put under investigation by gradually introducing noise into the different weight sets. In Figure 2, the effect of a full inversion of the causal weight sign, an inversion of the weights connecting positive regions and decisions (in that order), the full randomisation of the causal weight values and the randomisation of the weights connecting positive regions and decisions on the average kappa statistic. The results of this first experiment indicate an uneven contribution of different weight sets to the performance of the model.

Next, we investigate the effect of noise injection into these different weight sets according to the following system. We employ a *noise level*, increasing its value with 10 percentage point increments from 0 to 100 percent. Furthermore, the following logic applies regarding the implication of each noise level echelon:

- 0% - 50% : The sign of the weight is preserved, but the absolute value gradually decreases to zero with each increment.
- 50% : The value of the weight is equal to zero.

- 50% - 100% : The original sign of the weight is inverted and the absolute value is gradually increased to one with each increment.

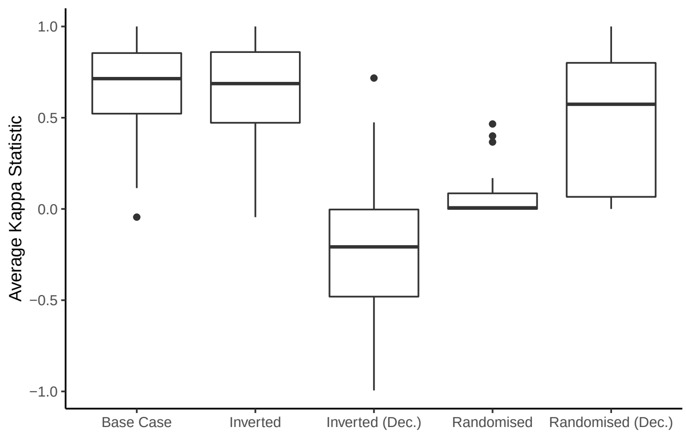


Figure 2: Average Kappa when the weight matrix is inverted or randomised

Figure 2 shows the results of the noise injection described above for the four weights sets described in (R_1^*) - (R_4^*). For each of these sets, no significant decreases in performance were observed when the sign was preserved. Sign inversion significantly deteriorates the model's performance, except for the cause-and-effect relation between positive regions. In this weight set, noise injection does not significantly affect performance.

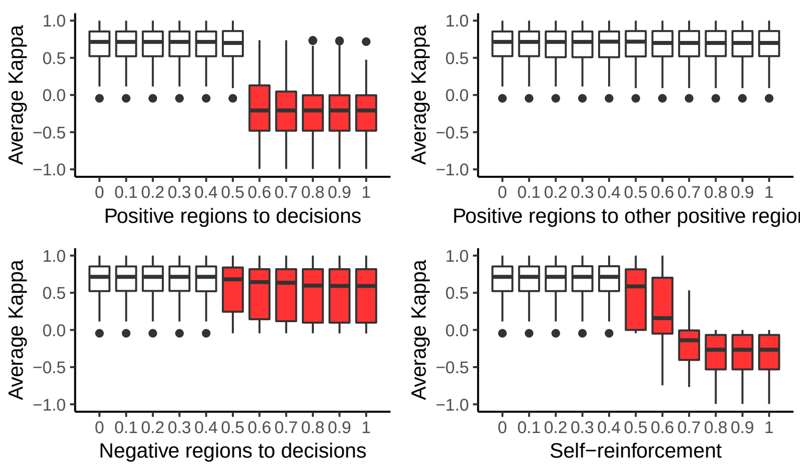


Figure 3: Effect of noise injection on the different weight sets in the weight matrix.

Table 2: Statistical evidence related to the noise injection in the different weight sets.

Noise Lvl.	p-value	Hypothesis	Noise Lvl.	p-value	Hypothesis
<50%	>0.26	Not Rejected	<50%	>0.13	Not Rejected
50%	0.7718	Not Rejected	50%	0.9652	Not Rejected
>50%	<1E-10	Rejected	>50%	>0.73	Not Rejected
Noise Lvl.	**p-value**	**Hypothesis**	**Noise Lvl.**	**p-value**	**Hypothesis**
<50%	>0.53	Not Rejected	<50%	>0.83	Not Rejected
50%	9.702E-06	Rejected	50%	9.702E-06	Rejected
>50%	<6.5E-06	Rejected	>50%	<2E-06	Rejected

The results shown in Figure 3 carry some interesting insights with respect to the model topology. Firstly, since adjusting the absolute value of the weights while preserving the original sign does not seem to affect performance, there is an indication that changing these weight values might not be as interesting as we presumed. Secondly, the fact that noise in connections between positive regions has no significant effect suggests that these connections might not be required to maintain model performance. If this statement can be confirmed by suppressing this connection, some pertinent questions could arise. Firstly, if we remove these connections, is the model still recurrent? Secondly, can we still call the underlying network a Fuzzy Cognitive Map? And finally, *why* are these connections not necessarily required to maintain performance?

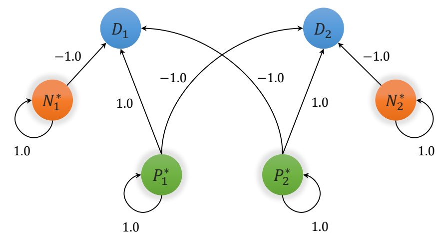
Table 3 shows that the suppression of these relations has a positive, yet statistically insignificant, impact on the model's performance (TL_1). The $\Delta kappa$ is defined by the following equation:

$$\Delta kappa = \begin{cases} 1 - \frac{\mathcal{K}}{\mathcal{K}^*} & , \mathcal{K}^* \geq \mathcal{K} \\ -1 + \frac{\mathcal{K}^*}{\mathcal{K}} & , \mathcal{K}^* < \mathcal{K} \end{cases} \tag{9}$$

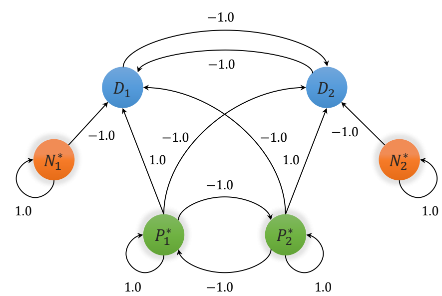
where $\mathcal{K}$ denotes the average kappa of the base case model and $\mathcal{K}^*$ denotes the average kappa of the alternative scenario.

Next to the suppression of positive region connections, we also explore the impact of other topological changes to the network. The first is a bilaterally negative connection between decision neurons (TL_2). The second is a connection from decision neurons to their respective positive region (with $w_{ij} = 1.0$) and positive regions belonging to other decisions (with $w_{ij} = -1.0$) (TL_3). In the third option, we investigated the effect of both of these changes when the relation between positive regions are suppressed (TL_4 - TL_5). Finally, we conducted a final experiment where we combine all aforementioned changes (TL_6). Figure 4 visualises each of these topological changes for a two-class classification problem.

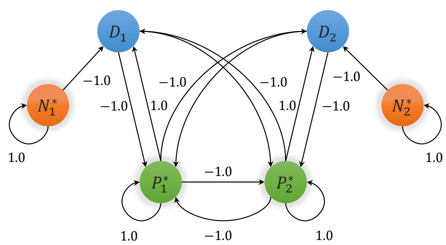
If we examine the results displayed in Table 3, we can see that topologies TL_1, TL_5 and TL_6 yield positive, yet statistically insignificant, results. These three alternative models have in common that the connection between positive neurons is suppressed. This means that even if these connections are suppressed, performance is not significantly affected. Furthermore, notice that TL_1 and TL_5 have the same outcome, despite TL_5 having a more complex topology. This reconfirms that more complex models are not necessarily more accurate.



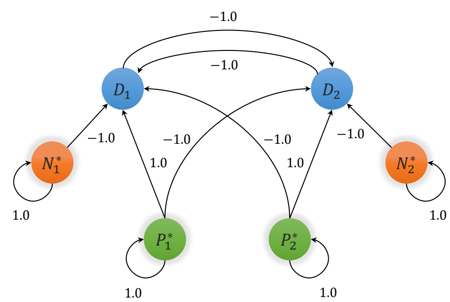
(a) TL_1



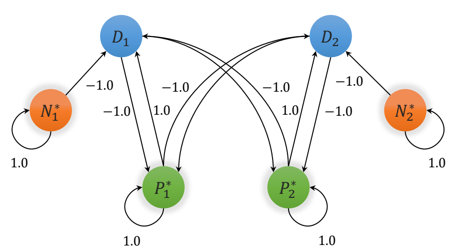
(b) TL_2



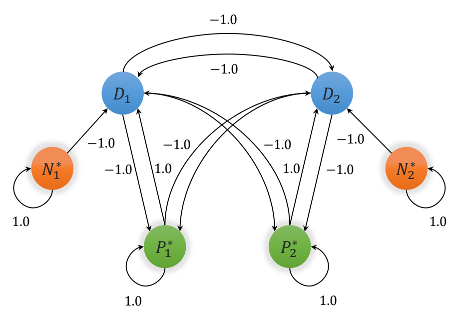
(c) TL_3



(d) TL_4



(e) TL_5



(f) TL_6

Figure 4: Changed topology scenarios TL_1 (positive region connections suppressed), TL_2 (decision connections added), TL_3 (connections between decision neurons and positive regions added), TL_4 (decision connections added and positive region connections suppressed), TL_5 (connections between decision neurons and positive regions added, positive region connections suppressed) and TL_6 (combination of all previous changes)

Table 3: Performance difference ($\Delta kappa$) per scenario.

Scenario	$\Delta kappa$	p-value	Holm	R^+	R^-	Hypothesis
TL_1	0.00888	0.03673	0.18365	18	9	Not Rejected
TL_2	-0.00025	0.7353	1.00000	3	3	Not Rejected
TL_3	0	1.00000	1.00000	0	0	Not Rejected
TL_4	-0.00025	0.7353	1.00000	3	3	Not Rejected
TL_5	0.00888	0.03673	0.18365	18	9	Not Rejected
TL_6	0.00937	0.0186	0.11160	16	7	Not Rejected

4.4. Activation Values

Introducing Gaussian noise into the initial activation values of input-type neurons allows us the assess (or rather re-confirm) the relevance of the initial state vector. Similar to the approach employed with the weight matrix, we use increasing noise levels, from 0 to 100 percent, with 10 percent increments. The null hypothesis is that introducing noise will significantly deteriorate an FRCN's performance. Figure 5 shows the results of this experiment.

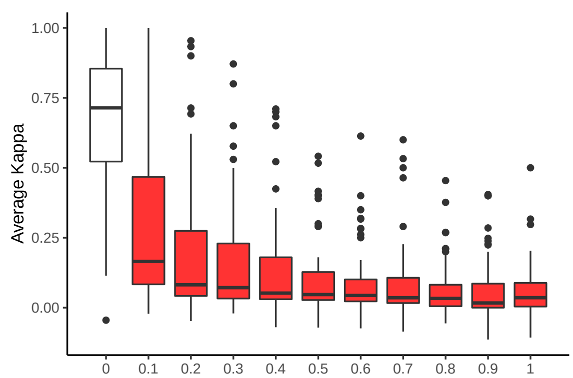


Figure 5: Effect of Gaussian noise injection in the initial activation values on performance.

Table 4: Statistical evidence related to the Gaussian noise injection in the activation values.

Noise Level	p-value	Holm	Hypothesis
10 %	2.278E-10	5.693E-10	Rejected
20 %	5.693E-11	5.693E-10	Rejected
30 %	6.017E-11	5.693E-10	Rejected
40 %	5.693E-11	5.693E-10	Rejected
50 %	6.358E-11	5.693E-10	Rejected
60 %	5.693E-11	5.693E-10	Rejected
70 %	6.358E-11	5.693E-10	Rejected
80 %	6.017E-11	5.693E-10	Rejected
80 %	6.357E-11	5.693E-10	Rejected
100 %	6.718E-11	5.693E-10	Rejected

As the results point out, we can reject the null hypothesis at all noise echelons. These results re-confirm the anticipated relevance of the activation state vector. As such, finding a more powerful way of activating input-type neurons is a scheme worth exploring. The correction of existing noise in the underlying data of the FRCN is an especially interesting research track.

4.5. Confidence Degrees

In Section 3, we already mentioned the integration of a confidence degree $\rho^{\Omega}_{X_k} \in [0, 1]$, derived from a different, potentially weaker classifier Ω. We described two possible approaches for this integration.

The first approach, providing a soft covering of the information space, entailed using $(R_5^{*'})$ and $(R_6^{*'})$ to compute the initial activation values of the positive and negative region neurons respectively. Figure 6 visualises this algorithmic change.

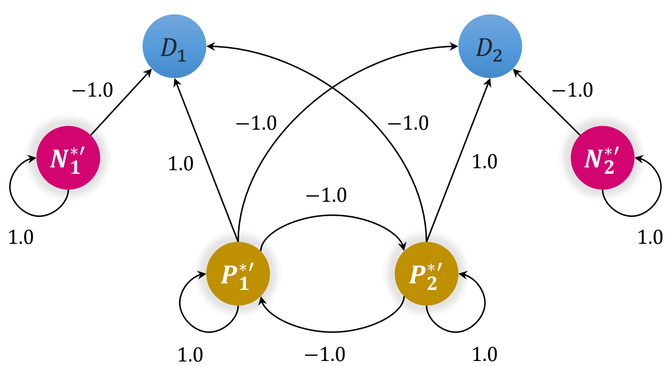


Figure 6: FRCN with activation values based on $(R_5^{*'})$ and $(R_6^{*'})$.

Table 5: Comparison of FRCNs with different Ωs.

Ω	$\Delta kappa$	**Holm**	R^+	R^-	**Hypothesis**
OneR	-0.16197	1.0000	9	39	Not Rejected
J48	-0.21534	1.0000	4	40	Not Rejected
BN	-0.28583	1.0000	7	45	Not Rejected
Log Reg	-0.22953	1.0000	9	41	Not Rejected
kNN	-0.21783	1.0000	4	42	Not Rejected

The second approach entails using Ω, in this case a white box classifier, to generate a confidence degree per decision class and subsequently using this degree to directly affect the initial state vector of the input-type neurons. Figure 7 visualises the topological change as a result of this second proposal.

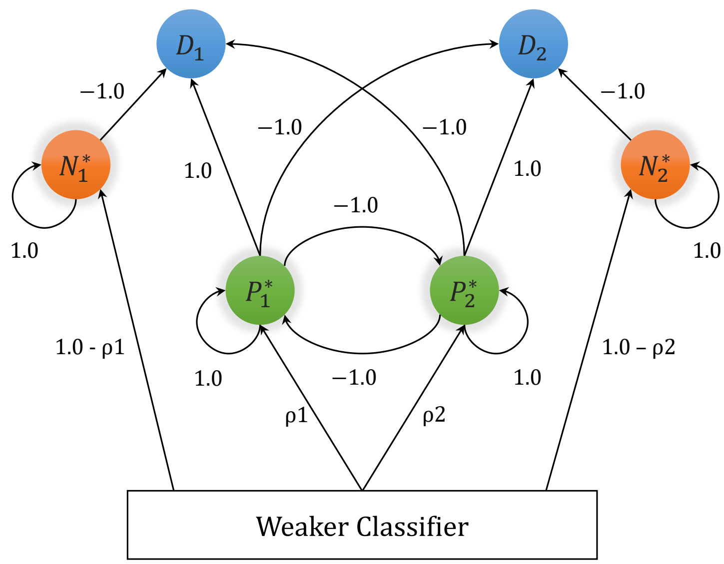


Figure 7: Hybrid version of an FRCN, with a white box classifier as Ω.

To determine the confidence weight of each positive and negative region in the network, the following rules apply:

- IF $C_i = P_k^*$ THEN $\rho_k = \rho_{X_k}^{\Omega}$
- IF $C_i = N_k^*$ THEN $\rho_k = 1 - \rho_{X_k}^{\Omega}$

where ρ_k is the confidence degree belonging to the k-th decision class.

Again, we employ the $\Delta kappa$ of the average kappa statistic, as described in Equation 9. To determine whether this change is positive and significant, we employ the Wilcoxon Signed-Rank Test with Holm correction.

Table 6: Comparison of the hybrid FRCNs with different Ωs.

Ω	$\Delta kappa$	**Holm**	R^+	R^-	**Hypothesis**
OneR	-0.31005	1.0000	10	43	Not Rejected
J48	0.00297	0.5739	25	23	Not Rejected
BN	0.01488	0.00055	34	12	Rejected
Log Reg	0.00128	0.29336	28	19	Not Rejected
kNN	0.00405	0.5739	23	24	Not Rejected

The results in Table 6 show that using Bayesian Networks significantly increases the performance of the FRCN, while all other options do not yield statistically significant results. This is interesting, as this would suggest that in some way, Bayesian Networks are able to produce different information than the FRCN and that this information leads to a correction of the results produced by the FRCN. Table 7 shows the number of instances where the original FRCN and Bayesian Network agree and disagree. The resulting $\Delta kappa$ using the hybrid model presented earlier is also included in this first table. The second table serves the same purpose, but compares consensus between the FRCN and the J48 Decision Tree.

Our hypothesis is that if a classifier reports more agreement with the FRCN, it will contribute less to the cooperative learning. As such, comparing the hybrid models with a Bayesian model, which produced positive results, and a J48 Decision Tree, which did not significantly affect performance, will provide an indication towards this last hypothesis. When comparing Tables 7 and 8, we can see that indeed, the Decision Tree's consensus degree is higher compared to the Bayesian Network. Therefore, there might be an indication that our hypothesis holds true. Yet, it is important to further verify this hypothesis in further research, with more classifiers and additional arguments.

Table 7: Consensus table for the FRCN and Bayesian Network.

ID	**Agree**	**Disagree**	$\Delta kappa$
1	120	0	0
2	120	0	0
3	874	24	-0.00572
4	866	32	0.03949
5	101	5	0.12669
6	198	28	0.02897
7	593	97	0.0284
8	159	46	-0.03479
9	468	157	0.02057
10	471	154	-0.01448
11	20	0	0
12	3919	1381	-0.00306
13	579	169	0.01639
14	224	62	-0.08531
15	0	198	0.01
16	80	25	0.00138
17	88	17	0.04305
18	1462	266	-0.00296
19	192	111	0.05123
20	316	52	0.016
21	283	65	-0.04442
22	489	11	-0.00221
23	21	3	0
24	748	725	0.05012
25	616	74	0.03493
26	609	81	0.04598
27	351	15	0.01038
28	124	8	-0.02662
29	282	54	0.0103
30	218	2	0.01599
31	218	2	0.02166
32	310	26	0.0103
33	319	17	0.0325
34	309	27	0.00803
35	265	71	0

Continued on next page

Table 7 – continued from previous page

Dataset ID	Agree	Disagree	$\Delta kappa$
36	399	369	0
37	419	349	0
38	427	309	0.06536
39	122	72	0.05357
40	159	55	0.04143
41	178	36	0.03505
42	199	15	0.08808
43	176	38	0.09428
44	199	15	0.03947
45	205	9	0.00632
46	134	80	0.01627
47	205	9	0.04143
48	149	65	0.02559
49	132	82	0.02559
50	30	276	0
51	142	8	0.01176
52	149	1	-0.01266
53	142	8	-0.01163
54	136	14	-0.03158
55	131	19	0

Table 8: Consensus table for the FRCN and J48 Decision Tree.

ID	Agree	Disagree	$\Delta kappa$
1	120	0	0
2	120	0	0
3	876	22	0.02414
4	830	68	0.05153
5	98	8	0.0279
6	165	61	0.09127
7	621	69	0.01318
8	158	47	0.04007
9	509	129	-0.06437
10	511	114	-0.10869
11	20	0	0
12	4917	387	-0.00271
13	620	128	-0.04432
14	238	48	-0.19695
15	146	52	0.02813
16	63	42	-0.05262
17	69	36	-0.00127
18	1620	108	0.01901
19	190	113	-0.03332
20	335	33	0.00553
21	287	81	-0.06411
22	499	1	0.00221
23	24	0	0
24	881	559	0.08714
25	639	51	0.00444
26	620	70	0.00887
27	337	29	0.0072
28	118	14	-0.10127

Continued on next page

Table 8 – continued from previous page

Dataset ID	Agree	Disagree	$\Delta kappa$
29	336	0	0
30	220	0	-0.00339
31	220	0	0.01542
32	336	0	0
33	308	28	0.04945
34	312	24	0.0414
35	260	76	0.08064
36	517	251	0.02305
37	499	269	0.00281
38	441	295	0.11281
39	121	73	-0.03579
40	153	61	-0.00705
41	170	44	0.02625
42	203	11	0.13914
43	164	50	0.12491
44	203	11	-0.01853
45	209	5	-0.0248
46	138	76	0.06514
47	209	5	-0.00705
48	138	76	-0.00017
49	120	94	-0.00017
50	55	251	-2
51	147	3	-0.0119
52	149	1	-0.08861
53	139	11	-0.05814
54	134	16	-0.01053
55	130	20	-0.01579

5. Concluding remarks

In this paper, we introduced the reader to Fuzzy-Rough Cognitive Networks. Furthermore, we investigated the contribution of the FRCN's building blocks to its performance and also highlighted the possible ways which might lead to an improvement of this performance. Specifically, we explored two main tracks to this end, the first being optimising the model's weight matrix and the second being implementing algorithmic changes based on either a soft covering of the information space or a hybrid approach.

The first track led to some interesting discoveries. The connections between the positive regions might not be necessary to maintain model performance. Further changes to the weight matrix, whether these changes entailed adding extra connections between neurons or changing the values of the weights, did not lead to different results either. Therefore, our hypothesis is that optimising the weight matrix by means of a learning algorithm, does not necessarily lead to an increased performance. Furthermore, the results showing that suppressing positive region connections might not affect performance provides us with additional research questions. Firstly, if we suppress these connections, can the model still be called a recurrent neural network? Secondly, can it still be called a Fuzzy Cognitive Map in that case? And finally, *why* would these connections not be necessary to maintain performance?

The second track also yielded some thought-provoking results. Using a white box classifier to produce a confidence degree per decision class, we implemented two different changes to the original algorithm. Firstly, we used the confidence degrees to transform the existing crisp fuzzy-rough environment to a softly covered one, which allows an object in the universe of discourse to belong to more than one similarity class at the same time. This approach did not yield positive results with respect to the performance. Secondly, we used the confidence degrees as weights, multiplying them with the initial activation values of input-type neurons. Here, we discovered that the performance of an FRCN can only be increased using a Bayesian Network as the white box classifier. The reason *why* this is the case, is an open question. The hypothesis is that the Bayesian Network produces additional insights from the available information in comparison to the FRCN, which allows the latter to correct wrongly labelled objects. We presented evidence to support this hypothesis using a consensus measure between the original FRCN and the Ω classifier, but these results should be expanded in further research.

References

[1] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, John Wiley & Sons, 2012.

[2] B. Kosko, Fuzzy cognitive maps, International Journal of Man-Machine Studies 24 (1).

[3] G. Nápoles, I. Grau, R. Falcon, R. Bello, K. Vanhoof, Rough cognitive networks, Knowledge-Based Systems.

[4] E. Papageorgiou, Review study on fuzzy cognitive maps and their applications during the last decade, in: IEEE International Conference on Fuzzy Systems, 2011.

[5] G. Nápoles, G. V. Houdt, M. Laghmouch, Q. Moesen, W. Goossens, B. Depaire, Fuzzy cognitive maps: a business intelligence discussion (2019).

[6] G. Nápoles, I. Grau, Deep learning channel, long-term cognitive networks and nonsynaptic backpropagation, IEEE Transactions On Systems, Man, And Cybernetics: Systems.

[7] Z. Pawlak, Rough sets, International Journal of Computer & Information Sciences 11 (1982) 341–356.

[8] G. Nápoles, Rough cognitive networks, Ph.D. thesis (2017).

[9] G. Nápoles, Rough cognitive ensembles, International Journal of Approximate Reasoning 85.

[10] G. Nápoles, C. Mosquera, R. Falcon, R. Bello, K. Vanhoof, Fuzzy-rough cognitive networks, Neural Networks.

[11] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, International Journal of General Systems 1.

[12] M. D. Cock, C. Cornelis, E. E. Kerre, Fuzzy rough sets: The forgotten step, IEEE Transactions on Fuzzy Systems 15 (2007) 121–129.

[13] R. Bello, R. Falcon, W. Pedrycz, J. Kacprzyk, Granular Computing: at the Junction of Rough Sets and Fuzzy Sets, Springer Verlag, 2008.

[14] E. I. Papageogriou, Learning algorithms for fuzzy cognitive maps - a review study, IEEE Transactions on Systems, Man, and Cybernetics 42 (2012) 150–163.

[15] L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.

[16] M. Inuiguchi, W.-Z. Mu, C. Cornelis, N. Verbiest, Fuzzy-Rough Hybridization, Springer Berlin Heidelberg, 2015.

[17] G. A. Papakostas, D. E. Koulouriotis, Classifying patterns using fuzzy cognitive maps, Springer Berlin Heidelberg, 2010, pp. 291–306.

[18] R. C. Holte, Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, Machine Learning 11 (1) (1993) 63–90. doi:10.1023/A:1022631118932.
URL https://doi.org/10.1023/A:1022631118932

[19] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.

[20] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian Network Classifiers, Machine Learning 29 (2) (1997) 131–163. doi:10.1023/A:1007465528199.
URL https://doi.org/10.1023/A:1007465528199

[21] D. W. Aha, D. Kibler, M. K. Albert, Instance-based learning algorithms, Machine Learning 6 (1) (1991) 37–66. doi:10.1007/BF00153759.
URL https://doi.org/10.1007/BF00153759

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The WEKA data mining software: an update, SIGKDD Explorations 11 (1) (2009) 10–18.

[23] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing (2017).
URL https://www.R-project.org/

[24] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (1945) 80–93.

[25] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (2) (1979) 65–70.

[26] M. Lichman, Uci machine learning repository (2013).
URL http://archive.ics.uci.edu/ml