

Reputation-Based Maintenance in Case-Based Reasoning

Nariman NAKHJIRI^{a,*}, Maria SALAMÓ^a, Miquel SÀNCHEZ-MARRÈ^b

^a*Facultat de Matemàtiques i Informàtica,
Universitat de Barcelona Institute of Complex Systems (UBICS),
Universitat de Barcelona (UB)*

^b*Knowledge Engineering and Machine Learning Group (KEMLG-UPC)
Intelligent Data Science and Artificial Intelligence Research Centre (IDEAI-UPC)
Dept. of Computer Science
Universitat Politècnica de Catalunya (UPC)*

Abstract

Case Base Maintenance algorithms update the contents of a case base in order to improve case-based reasoner performance. In this paper, we introduce a new case base maintenance method called Reputation-Based Maintenance (RBM) with the aim of increasing the classification accuracy of a Case-Based Reasoning system while reducing the size of its case base. The proposed RBM algorithm calculates a case property called *Reputation* for each member of the case base, the value of which reflects the competence of the related case. Based on this case property, several removal policies and maintenance methods have been designed, each focusing on different aspects of the case base maintenance. The performance of the RBM method was compared with well-known state-of-the-art algorithms. The tests were performed on 30 datasets selected from the UCI repository. The results show that the RBM method in all its variations achieves greater accuracy than a baseline CBR, while some variations significantly outperform the state-of-the-art methods. We particularly highlight the RBM_ACBR algorithm, which achieves the highest accuracy among the methods in the comparison to a statistically significant degree, and the RBM_{cr} algorithm, which increases the baseline accuracy while removing, on average, over half of the case base.

Keywords: Case-Based Reasoning, Case Base Maintenance, case property sets, case reputation

1. Introduction

In the field of Artificial Intelligence, Case-Based Reasoning (CBR) [1] refers to a problem-solving technique that follows a lazy learning strategy. The general

*Corresponding author: University of Barcelona, Spain; E-mail: nakhjna21@alumnes.ub.edu

methodology of a case-based reasoner is to solve new problems (i.e., also referred to as new cases) by retrieving the most relevant past problems from an existing knowledge-base (described in the field as a case base) and adapting them to fit new situations. Cases and their confirmed solutions are stored in a case base, which is a memory with a flat structure. Note that each case is defined as a set of attributes whose values describe a problem, as shown in Figure 1.

Aamodt and Plaza in [2] described the classical CBR model, which defines the problem solving cycle in four different phases. Figure 1 shows this CBR cycle, where a new case is solved by first *retrieving* one or more similar cases from the case base and then trying to *reuse* the information from them to provide a solution to the new case. Following, there is a *revision* of the proposed solution, and finally, the new case along with its confirmed solution can be *retained* to be used later by incorporating it into the case base. These phases are illustrated in Figure 1.

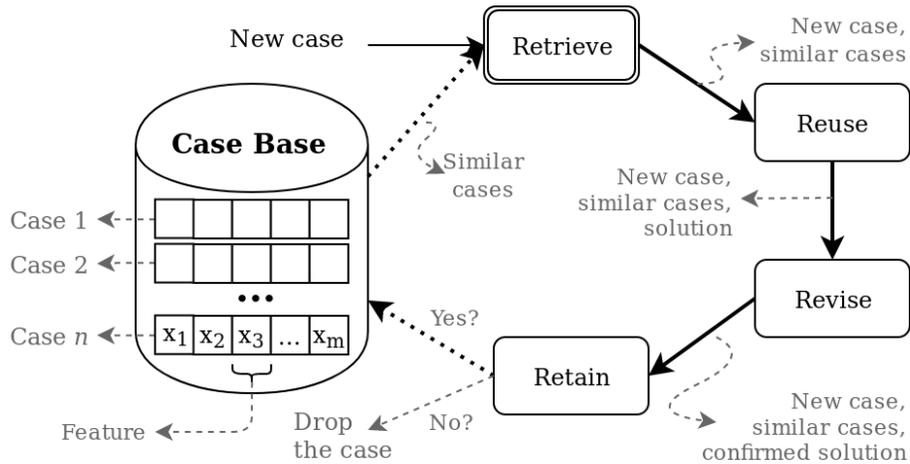


Figure 1: Case representation and CBR cycle

Since the introduction of CBR, numerous studies have been performed to enhance the performance of the technique. These improvements have been applied to different parts of the CBR process. While some studies have focused their attention on the reasoner part of the process, others have concentrated on the case base of the CBR. *Case Base Maintenance (CBM)* [3][4] as defined by Leake & Wilson refers to methods applied in CBR to manipulate and organize the content of its case base in order to improve or maintain the competence of the CBR technique. Maintenance algorithms are commonly divided into two categories: *Competence Enhancement* [5] and *Competence Preservation* [6] methods. *Competence Enhancement* refers to methods that identify and remove noisy and misleading information from the case base, while *Competence Preservation* are methods that aim to remove redundant cases from the case base without affecting the prediction accuracy. All the algorithms proposed in the *CBM* field are tagged with one or both of these categories. Both approaches

have their benefits. While an increase in accuracy is a common demand for a classifier, case base reduction has a direct effect on the performance of the CBR on large datasets. On the other hand, the computational time for *retrieving*
35 neighbors is decreased by reducing the number of cases stored in the case base.

In this work, we are concerned both with achieving an increase in accuracy and reducing the case base size in a CBR. We explore the benefits of incorporating a case property called *Reputation* into the CBR process. Our objective is to identify both useless and noisy cases based on their reputation. An additional
40 goal is to learn and reason from the CBR solving process and adapt the reputation of the cases in order to use it as a mechanism for improving maintenance. Thanks to the enrichment of the case base with the property of reputation, the CBR can remove appropriately those cases that cause harm while providing a quality case base with enhanced competence.

45 In this paper, our hypothesis is that Reputation improves both the case base size and the accuracy of a CBR. With this hypothesis in mind, the contribution of this paper is three-fold:

- First, we propose a Reputation-Based Maintenance (RBM) model.
- Second, we propose several variations of this model: RBM_{nr} , RBM_{cr} ,
50 RBM_{ining} , and RBM_ACBR .
- Third, we carried out an exhaustive evaluation of the proposals to validate our initial hypothesis.

Our evaluation focuses on a comparison of our proposals (i.e., RBM_{nr} , RBM_{cr} , RBM_{ining} , and RBM_ACBR) against baseline CBR and RENN,
55 BBNR, RDCL, and GCNN, which are well-known CBM strategies in the literature. The experiments were conducted on 30 datasets from the UCI [7] repository. With these experiments, we demonstrated the positive influence of the incorporation of the case reputation property into CBR. The results of our in-depth evaluation confirm our initial hypothesis and indicate that our proposals
60 improve significantly on the competence of previous algorithms in Case-Based Reasoning.

The rest of this paper is organized as follows. After a review of the related work in CBM in Section 2, we detail the RBM model and its variations in Section 3. Section 4 analyzes and compares the effectiveness of the proposals with the
65 base-line CBR and several well-known state-of-the-art CBM algorithms. Finally, in Section 5, conclusions are made along with directions for further research.

2. Related work

In the literature, several strategies have been defined for Case Base Maintenance. We categorize them as *direct models*, *hybrid models*, and *case property*
70 *models*. Most of the early attempts in CBM use direct models. Direct models use a competence model based on taking immediate action upon a classification of a case with a specific case base. These models do not use other techniques to

extract information on an individual case or on the relationships between cases. We will discuss these approaches and their development over time in Section 2.1. More recently are the hybrid and case property models. Hybrid models use different Artificial Intelligence techniques to outline certain relations among cases in the case base. On the other hand, *case property models* define sets and variables to capture the behaviour of individual cases on different scenarios. These models act on the case base when they have gathered enough evidence of a case performance. Section 2.2 reviews the studies of hybrid models, and Section 2.3 details the state-of-the-art in CBM methods with case property models.

2.1. Direct models

From the early attempts in CBM using direct models, we should mention *Hart's* Condensed Nearest Neighbors (CNN) [8], and *Wilson's* Edited Nearest Neighbors (ENN) [9]. CNN is a competence preservation [6] algorithm whereas ENN is in a competence enhancement [5] algorithm. Both CNN and ENN principles were the inspiration for many later algorithms in the CBM field.

CNN method is an incremental algorithm that starts with an empty case base. Cases are added to the case base if they are misclassified by the current members of it. *Selective Nearest Neighbors (SNN)* [10] and *Reduced Nearest Neighbors (RNN)* [11] are examples of methods that have reported improvements on CNN. Different *Instance-Based (IB)* [12] algorithms also use variations of the CNN technique. A more recent approach is Generalized Condensed Nearest Neighbor (GCNN) [13], in which the absorption condition of CNN is strengthened with GCNN by adding a threshold to the difference in distances of the closest class member and the nearest case to another class. Assigning 0 to the threshold results in a similar behaviour as with CNN.

On the other hand, ENN has also been used in different later research studies. ENN is a decremental algorithm that removes those cases from the case base that are misclassified by their k-nearest neighbors as it considers them as noise. Experiments show that the ENN model is successful at increasing the average accuracy of a CBR technique. Later on, *Tomek* proposed two methods based on ENN. In the first one, called *Repeated ENN (RENN)* [14], the ENN strategy is repeated on the case base until no further noises can be detected. The second method is *all kNN* [14], which also consists of performing several passes over the case base but on different neighborhood sizes ($\{1,2,\dots,k\}$ value in k-NN). Instances are removed if any value of k results in a misclassification. Another more recent variation on ENN was proposed by *Sanchez et al.* [15] using a k-nearest centroid neighbor (k-NCN) classifier rather than a nearest neighbor classifier.

2.2. Hybrid models

This category includes methods that build their models by integrating other artificial intelligence techniques. One example of hybrid models is *LSVM-nr* [16], which is based on *Local Support Vector Machine (LSVM)*[17][18]. In particular, the *LSVM-nr* algorithm trains a Support Vector Machine (SVM) on

the locality of each individual case in the case base. If for the tested case, the predicted probability of the actual class in its localized SVM is below a specified threshold, it would be considered noise, and thus removed from the case base. Training a LSVM on each instance in the case base can be very time consuming
120 when the size of the case base is large. For this reason, LSVM-nr was expanded by introducing *FaLKNR* [19]. *FaLKNR* achieves scalability by reducing the number of local SVMs that need to be retrieved and built while ensuring that the entire training set is covered by the models.

Another hybrid approach is the WCOID [20] algorithm, which uses the DB-SCAN [21] clustering method. WCOID is an extension of the earlier COID [22]
125 algorithm and its name refers to Weighting, Clustering, Outliers and Internal cases Detection.

2.3. Case property models

Case property models integrate additional information into the cases, called
130 properties, in order to better describe the characteristics of the cases on the case base. These properties are usually computed based on the performance of the cases in certain tasks or conditions.

The strategy of using case property sets to create a competence model was first used by *Smyth* and *Keane* in [23], where they introduced two important
135 case competence properties: *coverage* and *reachability* sets. The idea behind these properties is to monitor the behavior of the cases in the case base in order to estimate their competence. *Coverage* of a case $C \in CB$ has been defined as the set of all cases that C can successfully solve. On the other hand, the *Reachability* of a case C is the set of all cases that can successfully solve C . *Coverage* and *Reachability* have inspired many researchers to experiment with case prop-
140 erties. *McKenna & Smyth* [24] presented a family of competence-guided editing methods, based on the case properties of *Coverage* and *Reachability*.

The *Iterative Case Filtering (ICF)* [25] algorithm also uses the coverage and reachability properties of cases. ICF is a decremental strategy that focuses
145 on removing cases far from class borders. The *Blame-based Noise Reduction (BBNR)* [26] algorithm proposed by *Delany et al.*, defines another case property to build its competence model. This new property is called the *Liability* set of a case C , which is defined as the set of all cases that C causes them to be misclassified. This work was extended with another property set called
150 *Dissimilarity*. The four case properties of *Reachability*, *Dissimilarity*, *Coverage*, *Liability* were used in the *RDCL* [27] case base editing method. In *RDCL*, every case is categorized with a profile based on the contents of their property sets [28].

A more recent work is the *Competence Metric (CM)* method [29, 30], which
155 uses the case categorization of *McKenna & Smyth* [24] to evaluate cases in the case base. The metric used in the CM method is based on the size of the *Coverage* set of a case to its *Reachability* set. This metric helps to identify the cases to remove from the different categories. Another recent attempt with *Coverage* and *Reachability* property sets was by *Lu and Chang* in their *Closure*
160 [31] model. In the *Closure* method, these property sets are used to separate the

cases into different *Competence Groups* and *Competence Closures* that help to define a metric for the cases in the case base.

Instead of using common case properties, there are also models such as the *ACBR* [32] algorithm that defines another case property. Unlike other methods discussed here, *ACBR* does not go through the training case base to build its competence model, but it acts upon the entrance of new cases to the CBR. In *ACBR* competence model, cases are assigned with a *goodness* value. The goodness increases when a retrieved case *C* contributes towards a correct classification of another case, while if *C* attempts a misclassification its goodness drops. The *ACBR* removal policy selects cases from the case base with a *goodness* below a certain threshold. Moreover, [32] proposed different strategies for retaining new cases based on their *goodness*. There are other approaches based on the utility of a case, including (Sánchez-Marrè et al.[33]), which uses is a very similar concept to the goodness proposed in *ACBR*. The utility of cases starts at 0, and is increased or decreased, as the cases are used to solve problems. Periodically, all cases with a utility lower than a certain threshold are removed from the case base.

As we have described above, many proposals have been described in the literature. By their very nature, we have categorized them in *direct*, *hybrid* and *case property* models. In summary, *direct* models are, in general, fast and straight forward. Since there is no additional information required to build these competence models, they also use the least amount of memory compared to the other two approaches (i.e., *hybrid* and *case property* models). However, their performance is not as good as more complicated models. On the other hand, *hybrid* models are usually computationally more expensive since they use much more memory than *direct* and *case property* models. High performance is expected from *hybrid* models as a trade off for their computational cost. Finally, the *case property* models offer more balanced solutions regarding computational and memory cost and performance. Methods in *case property* models are capable of achieving better results in terms of performance than *direct* models while their computational cost is smaller than that of *hybrid* models. For this reason, in this paper we have focused on defining new *case property* models.

3. Reputation-Based Maintenance

Reputation-Based Maintenance (RBM) is an effective model for improving the performance of a CBR. The competence model of RBM is build on the *reputation* property of every case in the case base, which can be used in a variety of noise-removal or case-base reduction algorithms. Every variation of the RBM model uses the same process to calculate the reputation of every case in the case base. This common process is explained in Section 3.1. The reputation values reflect the competence of each case in the task of classification. In this paper, we propose four strategies for using the Reputation values in a CBM method.

This section consists of three major parts. Section 3.1 introduces the Reputation competence model. Section 3.2 describes two preprocessing of the CBM methods of the RBM model. Finally, in Section 3.3 the preprocessing CBM

205 methods of RBM are extended with learning capabilities and we introduce two new reputation-based strategies.

3.1. Reputation competence model

The reputation competence model is in charge of calculating the reputation value of each case in a case base. To measure the strength of cases in the classification task, RBM puts each case in the case base in the same scenario with a leave-one-out test and records their problem solving performance.

We want to see which cases are more likely to classify others correctly. Thus, in RBM the focus is on the cases' performance when they have been retrieved for a kNN classification.

215 *Reputation.* In RBM, the Reputation of a case C ($Rep(C)$) in the case base CB represent a value that shows its competence. Definition 1 shows the process of reputation calculation. $Rep(C)$ increases if C is in the neighborhood of another case $C' \in CB$ and belongs to the same class. When C and C' have different classes, $Rep(C)$ decreases. It is assumed that $\forall C \in CB, Rep(C) = 0$ as an initial state.

Definition 1. *Reputation*

$$Rep(C) : \begin{cases} +1, & \forall C' \in CB : \text{if } \{C \in \text{the } k\text{-neighborhood of } C' \\ & \wedge \text{class}(C) = \text{class}(C')\} \\ -1, & \forall C' \in CB : \text{if } \{C \in \text{the } k\text{-neighborhood of } C' \\ & \wedge \text{class}(C) \neq \text{class}(C')\} \end{cases}$$

At the end of the self test on the case base, every case has a reputation value based on its usage in the classification of other cases. If we separate the set of cases that have a positive impact on the $Rep(C)$ from those that have a negative effect, these two sets share a close resemblance to the Coverage ([23]) and Liability ([26]) sets. There is one difference between their definitions. While Coverage and Liability consider only the successful attempts in classification and misclassification, Reputation includes all the attempts regardless of the outcome of the kNN classification. When the value of k in kNN is set to 1, both approaches result in the same set, since the class of neighbor in 1NN is the same as the outcome of the class prediction. However, when $k > 1$, the set of cases with a positive impact on the $Rep(C)$ always includes the members of the $Coverage(C)$ set, along with cases from the same class as C , considering that C is in their neighborhood and they are misclassified. The same goes for the set of cases with a negative impact on $Rep(C)$ and $Liability(C)$. Before the maintenance process we do not know about the competence of the cases in the case base. The result of the classification is the outcome of k number of votes from a neighborhood and it might be altered by the noisy cases. So, the extension of Coverage and Liability sets by the reputation model makes RBM more tolerant to noisy cases in the case base, while Coverage and Liability are more sensitive regarding the case being tested. As RBM is a preprocessing

245 maintenance model that concentrates on the existing cases in the case base rather than new cases, the extension to the classic property sets of Coverage and Liability proposed by the RBM model will help the algorithm create a more competent case base.

Every variation of the RBM model uses this process to calculate the reputation of every case in the case base. The reputation value reflects the competence of each case in the task of classification. The next step in a maintenance algorithm is to organize the contents of the case base based on their case reputation values.

3.1.1. Reputation-based case categorization

In order to use the Reputation in a Case Base Maintenance method, it is first necessary to interpret its different values in the cases. Cases in the case base are categorized into three groups by their Reputation value:

255 **Definition 2.** *Reputation-based case categorization*

$$C \in CB : \begin{cases} Rep(C) \geq 1 & \text{group 1} \\ Rep(C) = 0 & \text{group 2} \\ Rep(C) \leq -1 & \text{group 3} \end{cases}$$

260 According to the Definition 2, the cases in the first group have positive Reputation values. Members of this group have more correct classifications in the record than misclassifications, and are the most valuable cases. The second group consist of the cases with a zero Reputation. Zero Reputation of a case is the result of not participating in any classification, or producing an equal number of right and wrong predictions. Finally, in the third group are those cases with negative Reputation values based on their poor performance in the classification of other cases. According to their record, they have produced more incorrect predictions than correct ones. Members of this group are the most likely to have a negative impact on the classification of other cases in the future.

3.2. RBM preprocessing CBM methods

270 This section describes the base model of RBM which branches out to RBM_{nr} in Section 3.2.1 as a Competence Enhancement method, and RBM_{cr} in Section 3.2.2 as a Competence Preservation method. A case's reputation is a reflection of its competence at classifying other cases. Additionally, to complete the maintenance process, a removal policy based on the obtained reputation values is defined. The basic scheme of the RBM model, as described in Algorithm 1, takes three input arguments: the case base CB , the neighborhood size of k , and the threshold of T . In particular, the $T \in \mathbb{N}$ value is used in the removal policy of the RBM. In the RBM model, cases find their place in the maintained case base (CB_{edited}) if their final reputation value is equal or greater than this threshold T . Based on different values of T , RBM model aims for different maintenance purposes, as we will see in Section 3.2.1 and Section 3.2.2.

Algorithm 1 *RBM*

```
1: function RBM(CB, k, T)
2:   for  $C \in CB$  do
3:      $Rep(C) = 0$ 
4:   for  $C \in CB$  do ▷ Leave one out
5:      $Neighbors(C, k) \leftarrow$  k nearest neighbors of C
6:     for  $C' \in Neighbors(C, k)$  do
7:       if  $class(C) == class(C')$  then
8:          $Rep(C') = Rep(C') + 1$ 
9:       else
10:         $Rep(C') = Rep(C') - 1$ 
11:     $CB_{edited} = \{\}$ 
12:    for  $C \in CB$  do
13:      if  $Rep(C) \geq T$  then
14:         $CB_{edited} = CB_{edited} \cup \{C\}$ 
15: return edited_CB
```

280 As show in Algorithm 1, the maintenance process consists of three main loops starting at lines 2, 4, and 12. The first loop at line 2 is an initialization to assign the value 0 for the reputation of every case in the case base ($\forall C \in CB$). The second loop at line 4 is the leave-one-out test on the case base *CB* to calculate the reputation of each case. The process, as explained in Definition 1, begins
285 with the selection of the *k* nearest neighbors of a case *C*. The neighboring cases are stored in the list $Neighbors(C, k)$ (line 5). The inner loop of the leave-one-out cycle starting at line 6 checks the class of each neighbor (*C'*) with the test subject to adjust its reputation ($Rep(C')$). By assigning an empty set for the edited case base, the process enters the final loop at line 12. This is where the
290 removal policy applies to the initial case base. *RBM* adds every case from *CB* that has an equal or higher reputation value than the predefined threshold *T* ($Rep(C) \geq T$), to the edited case base (CB_{edited}). Finally the algorithm finishes by returning the CB_{edited} as the maintained case base.

With the general form of the RBM, we propose two methods to achieve differ-
295 ent goals. RBM_{nr} is the first method that defines a Competence Enhancement method, aimed at removing noisy cases from the case base (Section 3.2.1). The second proposed method is the RBM_{cr} , a Competence Preservation method aimed at reducing the size of the case base as much as possible while maintaining the accuracy (Section 3.2.2). These are the base preprocessing methods of
300 the RBM, which are later expanded with learning in Section 3.3.

3.2.1. RBM_{nr}

In RBM_{nr} where *nr* stands for noise removal, the goal is to detect and remove harmful cases from the case base. With the Reputation model, cases are categorized into three groups as explained in Section 3.1.1. The third group
305 according to the Definition 2 consists of those cases with a negative Reputation value, which show their negative effect on the classification of other cases. Cases

with a negative Reputation ($Rep(C) < 0$) according to the other members in the case base has produced more false predictions than correct ones. Thus, it is more likely to give wrong predictions in the future and this is harmful for case-based reasoning technique. Using the RBM scheme presented in Algorithm 1, RBM_{nr} is called by replacing the input argument of T for the threshold established at 0. Definition 3 shows the relationship with the RBM scheme, when RBM_{nr} is called for a case base CB with the neighborhood size of k .

Definition 3. RBM_{nr}

$$RBM_{nr} \equiv RBM(CB, k, 0)$$

In the RBM_{nr} , we keep those cases with a zero Reputation value in the maintained case base. In a noise removal strategy, cases are removed from the case base if we have evidence of their harmful overall performance. In the scenario of cases with zero Reputation value, the method does not have enough information to determine its reliability in the future. For this reason, the cases with a zero Reputation have not been identified as noise.

3.2.2. RBM_{cr}

In RBM_{cr} , where cr stands for case base reduction, the goal is to define a competence preservation method. A method in Competence Preservation, should significantly decrease the size of the case base while maintaining its accuracy. This goal is achieved by removing harmful, redundant and useless cases from the case base. To identify those cases with the Reputation model we refer to the case categorization described by Definition 2. Cases with a negative Reputation are the first group of cases that are deleted from the case base. Removing these harmful cases reduces the size of the case base and increases accuracy. For a case with a reputation value equal to zero, if the outcome is the result of that case not being used in any classification, it indicates that the case is irrelevant for the classification of other cases according to the other cases in the case base. These useless cases are removed from the case base with a minimum effect on accuracy. The remaining cases with a reputation value equal to zero consists of those cases that have an equal number of correct and incorrect predictions. These borderline cases can become noise or valuable cases depending on their next performance, but their current state indicates an unreliable performance. They could contribute to increasing the accuracy or decreasing it. Removing them from the case base therefore serve to maintain accuracy at the same level. In RBM_{cr} we remove all the cases with a negative or zero Reputation and preserve only those with a positive Reputation value.

Definition 4. RBM_{cr}

$$RBM_{cr} \equiv RBM(CB, k, 1)$$

Definition 4 shows the RBM_{cr} setup using the RBM scheme detailed in Algorithm 1. On a case base CB with the neighborhood size of k , RBM_{cr} replace the threshold of T with the value 1.

3.3. Extended models with learning

Maintaining the case base in a case-based reasoner takes place at various steps in the process. The reputation-based models we discussed in Section 3.2 preprocess the CBM methods that clean the case base before the CBR is exposed to new cases. Another place where maintenance process happens is after the revision of new cases, at the retention phase. Retention strategies allow the CBR to learn and keep their case base constantly updated as new cases are learned according to a criterion. In this section we propose two extended CBM methods based on the previous ones that also include a learning strategy. Section 3.3.1 defines an algorithm that uses RBM in a Case Base Mining setup and Section 3.3.2 describes a method combining RBM and ACBR [32].

3.3.1. RBMining

In case base maintenance, if we remove a case there is no way to return it to the case base. This issue has been covered in case base mining, where the case base used for classification will always be obtained from all the data. Case Base Mining is defined by *Pan* in [34], as a maintenance algorithm that takes input from a raw case set, whereas case base maintenance takes input from an existing case base to improve its performance. This significantly reduces the effect of order and time on the result of the maintained case base.

According to *Pan*'s research there are two major problems in previous case-mining algorithms. The first problem is caused by noisy cases and the second problem is unbalanced data and uneven data distribution. In both of these situations some unwanted cases are removed. Noisy cases cause other cases in the case base, which would benefit the classification of future problems, to be removed along with them. Such situations are usually the result of a lack of knowledge about the true distribution of data. In a case base maintenance method, the competence of a case is dependent on the current state of the case base. A case that has been identified as incompetent its current situation may be a valuable member if we obtain more knowledge about its neighborhood in future problems. In case base mining algorithms, there are two memories (case bases) devoted to maintaining the cases used in the CBR technique. The first one is called Memory Bank (MB), and it is a place to store all past and upcoming cases. The second is Active Memory (AM), which is a maintained case base for the classification of new problems. In the learning process, AM is updated not on itself but on MB. A problem in MB will always have a chance to prove its competence and find its way back to AM. One of the downsides of case base mining is its memory consumption and the amount of time it needs for training its AM. Active Memory updates its content based on MB whenever necessary. Separating MB and AM gives the maintenance process a chance to work to its full potential. We consider that increasing the number of cases in the MB whenever a new case has been learnt during the retention phase to the CBR, results in a better Reputation-Based model. Bringing more evidence to the reputation of a case increases the reliability of the model and should improve its general accuracy overtime.

Algorithm 2 $RBMining_{pre}$

```
1: function  $RBMining_{pre}(MB, k)$  ▷ MB is the Memory Bank  
2:    $AM = \{\}$  ▷ Initiating Active Memory (AM)  
3:    $AM = RBM(MB, k, 0)$   
4: return  $AM$ 
```

390 The implementation of case base mining for RBM is divided into two parts.
First part takes place at the preprocessing stage and the second is performed
during the retention phase. *Algorithm 2* shows the first stage in RBM case-base
mining ($RBMining$) during preprocessing. For testing purposes, the method
of RBM_{nr} has been selected as the base algorithm of *Reputation-Based Mining*
395 ($RBMining$).

According to *Algorithm 2*, the preprocessing step of the $RBMining$ method
is quite similar to the basic form of RBM_{nr} . The difference is in the separation
of memories. The $RBMining_{pre}$ function receives the MB as an input and
returns the AM. *Reputation* value calculations are performed on all the cases in
400 the MB. Accordingly, instances with a reputation value of zero and above will
then be added to the AM. No changes are made to the cases in the MB and,
they therefore remain intact.

Algorithm 3 $RBMining_{ret}$

```
1: function  $RBMining_{ret}(MB, k, C_{test})$   
2:    $MB = MB \cup \{C_{test}\}$   
3:    $AM = \{\}$   
4:    $AM = RBMining_{pre}(MB, k)$   
5: return  $AM$ 
```

Moving on to the retention phase of the $RBMining$ algorithm, new cases like
 C_{test} which have passed the revision step of the CBR will be added to the MB
405 unconditionally. But the C_{test} should hold a non-negative reputation in the MB
($Rep(C_{test}) \geq 0$) to find its way to the AM. In *Algorithm 3*, The $RBMining_{ret}$
function has been presented to perform the procedure explained. Procedures
are described with an easy to read pseudo-code.

3.3.2. RBM_ACBR

410 There is a close relationship between *Reputation* in RBM and *Goodness* in
 $ACBR$ [32]. Both of these competence values change in a similar pattern. In the
k-NN classification of new cases, retrieved neighbors in RBM and $ACBR$ change
their competence. A case $C' \in CB$ that belongs to the k-nearest neighbors of the
case to be classified, C , raises its *goodness* or *Reputation* if it has the same class
415 as C . On the other hand, the *Goodness* or *Reputation* value of C' drops when it
has any other class than C . This similar behavior is the result of same rationale
regarding case interactions. The main difference between ACBR and RBM is
the stage of CBR at which they take place. While RBM is a preprocessing

case base maintenance operation, ACBR manipulates the case base during the
420 problem solving process and learns new cases during the retention phase. The
decision to maintain the case base during the classification of new cases has its
own up and down sides. The upside is that the process of maintenance is much
faster in ACBR as there is no leave-one-out test in the preprocessing stage.
Additionally, no extra calculation are made for *Goodness* changes other than
425 the necessary steps in a normal CBR and k-NN classification. However, the
downside is the slowness of changes in the case base. In order to see the full
effect of the *Goodness* competence model, CBR has to wait for numerous new
problems to be solved to build its desired case base. Another problem is the
sensitivity to the order in data appearance, especially for newly absorbed cases.

430 On the other hand, RBM utilizes the pattern of *Goodness* changes and puts
it in a preprocessing step. Although the maintenance process becomes much
slower than with ACBR, *RBM* is much more robust against order sensitivity.
Additionally, in RBM we see immediate changes to the case base as we use
all the cases in the training case base to manipulate *Reputations*. Due to the
435 similarities of *ACBR* and *RBM* in case evaluation, and considering they take
place during different phases of the CBR, we proposed a combined method called
RBM_ACBR. This is an algorithm that performs *RBM_{nr}* as a preprocessing
maintenance step and uses *ACBR* for its learning and continuous maintenance
strategy. *RBM_{nr}* was selected as the base method of preprocessing because
440 *ACBR* is also a noise reduction technique. This means that both parts of the
combined method work to the same purpose and the CBR is more coherent.

4. Evaluation

In this section, the performance of the proposals is assessed and compared
to well-known state-of-the-art CBM methods and with a plain CBR.

4.1. Data description and Methodology

445 The evaluation was performed using 30 data sets with different character-
istics from the UCI machine learning repository [7]. The selected data sets
were chosen to provide a varied number of cases, features, classes, and different
proportions of nominal and numerical attributes. Additionally, some of them
450 contained a significant percentage of missing values. The details of the data
sets are briefly described in Table 1. By comparing the two numbers in the last
column, we can conclude whether a data set had a balanced or unbalanced class
distribution among its population.

	Dataset	Dataset size	# Num	# Nom	missing values	# Classes	class ratio (Most/Least)
AD	audiology	226	0	69	Yes	24	25.2% / 0.4%
AT	autos	205	15	10	Yes	6	32.7% / 1.5%
BL	bal	625	4	0	No	3	46.1% / 7.8%
BI	biopsies	1027	24	0	No	2	51.6% / 48.4%
BR	breast-w	699	9	0	Yes	2	65.5% / 34.5%
BP	bupa	344	6	0	No	2	58.0% / 42.0%
CI	cicyt	648	4	2	No	5	45.8% / 1.5%
CM	cmc	1473	2	7	No	3	42.7% / 22.6%
CO	colic	368	7	15	Yes	2	63.0% / 37.0%
CR	credit-a	690	6	9	Yes	2	55.5% / 44.5%
FI	fis	216	21	0	No	2	56.0% / 44.0%
GL	glass	214	9	0	No	6	35.5% / 4.2%
GR	grid	1888	2	0	No	2	50.0% / 50.0%
HC	heart-c	303	6	7	Yes	2	54.5% / 45.5%
HH	heart-h	294	6	7	Yes	2	63.9% / 36.1%
HS	heart-s	270	13	0	No	2	55.6% / 44.4%
HP	hepatitis	155	6	13	Yes	2	79.4% / 20.6%
IO	ionosphere	351	34	0	No	2	64.1% / 35.9%
IR	iris	150	4	0	No	3	33.3% / 33.3%
LB	labor	57	8	8	Yes	2	64.9% / 35.1%
LY	lymph	148	3	15	No	4	54.7% / 1.4%
MX	mx	2048	0	11	No	2	50.0% / 50.0%
PI	pima-i	766	8	0	No	2	65.1% / 34.9%
PT	primary-t	339	0	17	Yes	21	24.8% / 0.3%
SG	segment	2310	19	0	No	7	14.3% / 14.3%
SN	sonar	208	60	0	No	2	53.4% / 46.6%
SB	soybean	683	0	35	Yes	19	13.5% / 1.2%
VE	vehicle	846	18	0	No	4	25.8% / 23.5%
WI	wine	178	13	0	No	3	39.9% / 27.0%
ZO	zoo	101	1	16	No	7	40.6% / 4.0%

Table 1: Datasets details: The columns in the table sequentially show: (1)acronym of the dataset, (2)full name of dataset, (3)number of cases, (4)number of numerical features, (5)number of nominal features, (6)presence of missing values, (7) number of classes, and (8) class population ratio for the most crowded class and the rarest one.

455 In order to evaluate the RBM model, we analyzed the different options we
have presented so far: RBM_{nr} , RBM_{cr} , RBM_{ining} , and RBM_{ACBR} . We
first compared the four different model configurations against the standard CBR
method. Additionally, we compared the performance of our model against that
of well-known state-of-the-art CBM algorithms. In particular, we chose RENN
460 [14], BBNR [26], RDCL [27], and GCNN [13] methods for our comparison.
RENN and GCNN are *direct* models and were selected for the evaluation be-
cause RENN is one of the most used and well-known methods in the literature
and GCNN is a recent development in this category. The BBNR and RDCL
methods, have been selected because they belong to the same category as RBM.
465 They are well-known methods for *case property* models and share most simi-
larities with RBM. Note that RBM model does not use any other artificial in-
telligence technique to build the reputation model. For this reason, it is out of

the scope of this paper to evaluate our proposals against *hybrid* models because they build their models by integrating other artificial intelligence techniques, requiring a lot of memory and being computationally expensive.

To perform a fair comparison, all the tests were performed using the same configuration. Every algorithm is tested on each data set using a 10-fold cross validation. The CBR, integrated in every CBM method, was performed based on a Euclidean metric as a distance measure and *1NN* as the size of the neighborhood in the retrieval phase. Accuracy and the case-base reduction of each tested algorithms were measured during the tests. Note that classification accuracy (i.e., the average of correctly classified cases) is the most common performance measure used in CBR.

4.2. Comparison of the RBM model configurations against CBR

In order to establish a baseline for the evaluation of RBM, the results of a plain CBR technique without any maintenance methods (noted as the *CBR* in the evaluation) were considered. Table 2 shows the outcome of the tests on a plain CBR and the proposed CBM methods. The first column in Table 2 shows the dataset identifier (Data id). The second column depicts the results of a plain CBR. The third column corresponds to RBM_{nr} . The fourth and fifth columns show the results of RBM_{cr} with 1 and 3 neighbors, respectively. Column 6 and 7 correspond to RBM_{mining} with 3 neighbors and a *nr* setup and 1 neighbor and a *cr* setup, respectively. Finally, the last column shows the results obtained with a combined approach of RBM and ACBR, called here RBM_ACBR .

The last two rows in Table 2 depict, 'Acc', representing the average accuracy, and 'CBred', which shows the average case base reduction of the tested methods for all the datasets.

The observations made from the results depicted in Table 2 are set out in depth in Sections 4.3 and 4.4.

4.3. Preprocessing RBM evaluation

Section 4.3.1 details the performance of RBM_{nr} (RBM(CB,k,0)), and Section 4.3.2 is focused on the analysis of RBM_{cr} (RBM(CB,k,1)) outcomes.

4.3.1. RBM_{nr} performance

RBM_{nr} is the competence enhancement variation of the RBM model for preprocessing maintenance. According to the tests shown in Table 2, RBM_{nr} 's best performance is achieved with the $RBM_{nr}(CB, 3, 0)$ configuration, which indicates that the RBM works on the case base CB with a neighborhood size of 3, and in noise removal (nr) mode¹. Note that the accuracy is on average 79.10 whereas the plain CBR obtains 77.24. Figure 2 puts the obtained results from different RBM_{nr} 's configurations into perspective by comparing them with

¹It is worth mentioning that the neighborhood size of the RBM model is independent of the neighborhood size of the classifier in the reasoner part of the CBR. In other words, the classifier can still be run by a 1NN, while the maintenance policy uses 3NN.

Data id	CBR	RBM (CB,3,0)	RBM (CB,1,1)	RBM (CB,3,1)	RBMin- (CB,3,0)	RBMin- (CB,1,1)	RBM _ACBR
AD	75.36	74.59	67.56	67.76	73.59	67.14	75.03
AT	74.63	66.67	65.18	66.42	68.16	64.77	68.12
BL	76.16	84.32	85.59	85.78	83.84	85.59	84.32
BI	83.18	83.28	81.05	82.04	84.43	81.03	83.57
BR	95.86	96.26	96.25	96.55	96.41	96.54	96.41
BP	62.93	64.24	62.87	63.92	64.56	62.30	65.40
CI	61.53	60.71	60.78	62.33	60.61	61.38	61.36
CM	44.40	45.27	45.54	46.01	45.68	46.08	45.34
CO	73.36	84.74	83.44	83.68	84.74	83.46	85.02
CR	81.76	86.67	85.07	85.95	86.53	85.21	86.52
FI	63.93	64.46	64.22	63.48	64.44	63.87	62.96
GL	66.31	65.55	67.38	64.57	66.45	67.83	67.83
GR	96.13	95.81	95.39	95.71	95.97	95.34	95.97
HC	74.20	78.83	79.81	77.54	79.49	80.88	79.16
HH	72.83	76.60	77.66	80.01	76.90	78.31	76.95
HS	74.07	77.78	79.63	77.41	78.89	80.37	77.78
HP	78.00	81.38	80.63	85.10	80.71	79.92	81.38
IO	86.93	88.89	89.16	89.43	88.32	88.89	88.89
IR	95.33	95.33	96.00	95.33	95.33	96.00	95.33
LB	83.38	88.24	82.81	82.57	88.24	82.81	88.24
LY	83.28	82.22	80.78	80.16	82.22	81.22	82.22
MX	78.61	78.31	76.36	78.02	78.36	76.36	78.31
PI	70.73	76.86	74.92	75.82	76.47	75.05	76.47
PT	38.45	46.77	42.29	44.28	45.89	43.19	46.47
SG	97.36	96.97	95.84	95.71	97.01	95.93	97.01
SN	86.84	84.51	80.72	83.51	84.94	81.63	84.92
SB	82.15	88.94	88.03	89.81	88.95	88.05	88.79
VE	69.44	69.57	67.34	69.45	68.60	67.32	69.32
WI	95.64	95.64	94.46	96.16	95.64	94.46	95.64
ZO	94.63	93.52	91.68	93.56	92.52	91.68	94.52
Acc	77.24	79.10	77.94	78.60	79.13	78.08	79.31
CBred	0.00	16.26	55.82	36.66	6.68	50.99	14.03

Table 2: Accuracy and case base reduction of plain CBR and RBM models.

well-known state-of-the-art CBM methods. The left-hand side of Figure 2 shows accuracy, and the right side depicts case base reduction.

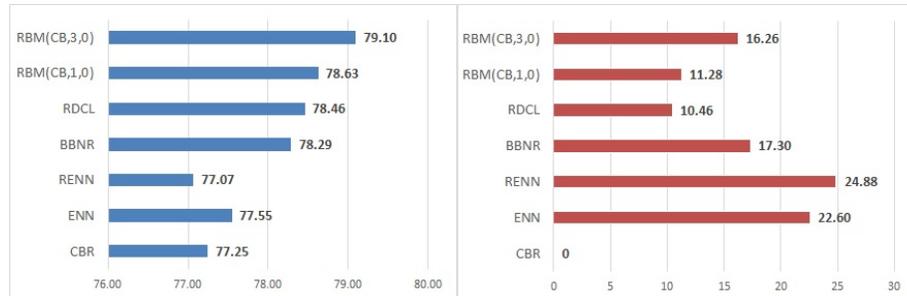


Figure 2: RBM_{nr} performance versus state-of-the-art methods

On average, the RBM_{nr} improves the accuracy of a CBR technique. Both $RBM(CB, 3, 0)$ and $RBM(CB, 1, 0)$, outperform CBR. In addition, the accuracy improvement over CBR achieved by RBM_{nr} is greater than all the compared state-of-the-art methods in the test. However, not all the datasets achieved greater accuracy when they underwent the RBM_{nr} maintenance process. Out of 30 datasets in comparison to baseline, $RBM(CB, 3, 0)$ managed to improve or maintain the accuracy of 20 of them. Although the RBM_{nr} is a competence enhancement method, it also reduces the size of the case base as a side effect of removing noisy cases. In the case base reduction category, RBM_{nr} ranks among the average of its peers and it removed 16.26 percent of the case base.

Another evaluation mode has been on analyzing the impact of the different neighborhood sizes on the performance of the RBM_{nr} . As it is shown in the Figure 2, RBM_{nr} with the neighborhood size of 3, surpass its other configurations in both average accuracy and case base reduction. This result shows that by expanding the effective neighborhood in *Reputation* calculation model has improved its performance.

4.3.2. RBM_{cr} performance

The competence preservation variation of the preprocessing CBM methods in the RBM model is the RBM_{cr} . The main focus of the RBM_{cr} is on reducing the case base with the least damage possible to accuracy. Although the methods in this paper, that the RBM_{cr} has been compared to, are in the competence enhancement category, the proposed method achieved higher accuracy than the plain CBR and some of the state-of-the-art methods, as shown in Table 2. This achievement is even more significant when the case base reduction is also taken into account. RBM_{cr} removes more than half of the case base (55.82% on average) and is still able to keep the classification competence higher than the CBR, as shown in Table 2. Figure 3 shows the case base reduction by the RBM_{cr} for every datasets in the test.

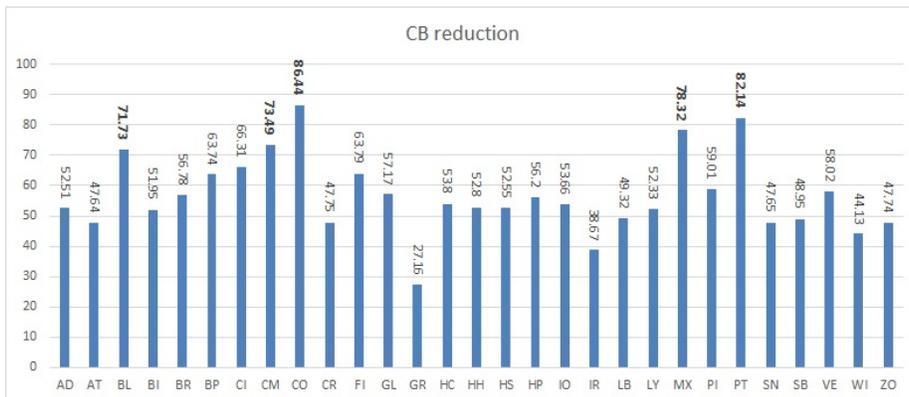


Figure 3: Case base reduction of $RBM_{cr}(CB, 1, 1)$ for all datasets

According to Figure 3, the most significant performance of the RBM_{cr} was in datasets that had reduced more than 70% of their case base. There were 5 such data sets in the test. The first one is the *bal* (*BL*) data set, with a 71% case base reduction, which experienced an improvement in accuracy of 9.43% in comparison to the CBR. Another such dataset is *cmc* (*CM*), which, with a reduction of 73% of the case base, still managed to score 1.1% better accuracy than the CBR. The largest cut was on the *colic* (*CO*) data set which again has achieved 10.08% higher accuracy than CBR. The situation is the same with the *primary-tumor* (*PT*) and the *MX* was the only dataset in which we observed a small drop (2.25%) in accuracy when compared to the CBR.

To obtain a better perspective of the RBM_{cr} performance, the competence preservation methods of CNN [8] and GCNN [13] were also implemented for the evaluations. Figure 4 shows the summary of the test results for the performance comparison of RBM_{cr} against plain CBR and well-known state-of-the-art CBM methods in both the competence enhancement and competence preservation categories. Again, the left-hand side of Figure 4 shows the accuracy and right-hand side depicts the case base reduction.

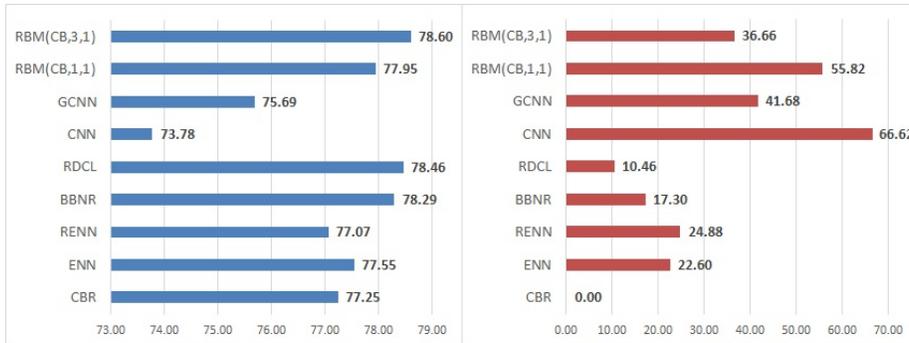


Figure 4: RBM_{cr} performance versus state-of-the-art

It is clear from Figure 4 that the proposed methods outperform the GCNN by a large margin. CNN however, has a higher case base reduction rate but reduces the accuracy of the case base significantly. According to Figure 4, compared to the rest of the methods, while the RBM_{cr} with a neighborhood size of one have a higher accuracy than the CBR and some of the state-of-the-art methods, RBM_{cr} with a neighborhood size of 3 outperforms the accuracy of all the methods in its scope. The last step in the evaluation of the RBM_{cr} was to compare RBM model performance with different neighborhood sizes. Following the same pattern as the RBM_{nr} in accuracy, by increasing the size of the neighborhood in the preprocessing, the average accuracy increased. Increasing the neighborhood size in RBM_{cr} had the opposite effect on its case base reduction compared to RBM_{nr} . While RBM_{nr} obtains a greater case base reduction by increasing the neighborhood size, $RBM(CB, 3, 1)$ had a smaller case base reduction than $RBM(CB, 1, 1)$.

4.4. Learning RBM evaluation

The main task of a learning algorithm is to preserve and improve quality over time in a scenario in which the system is exposed to learning new data. The learning methods of RBM are no exception, as both *RBMining* and *RBM_ACBR* improve their performance not only in comparison with the plain CBR, but also with regard to their related preprocessing methods, as shown in Table 2. Section 4.4.1, details the evaluation of *RBMining* and Section 4.4.2 describes the performance of the *RBM_ACBR*.

4.4.1. RBMining performance

In the case base mining setup, the RBM model does not remove any of the previous cases from its Memory Bank and always adds new cases to it. The result of adding more information in the construction of RBM model is positive according to the results depicted in Table 2. With the noise removal setup, *RBMining* achieves the second highest average accuracy over the 30 data sets, just after the other learning method of the RBM model (*RBM_ACBR*). The classification accuracy of the *RBMining* in both 'nr' and 'cr' mode is above the state-of-the-art methods analyzed in this paper. On the case base reduction front, an important point to keep in mind is that a learning method in the 10-fold cross validation setup can add up to 10 percent of the initial population to the case base size. Therefore, learning methods in the test have a lower case removal rate than their preprocessing counterparts. In *RBMining* since the learning algorithm is the same as the preprocessing one, we notice no significant changes in the case base reduction rate compared to the preprocessing CB methods ².

Config	$(CB, 1, 0)$		$(CB, 3, 0)$		$(CB, 1, 1)$		$(CB, 3, 1)$	
Method	RBM	RBMIn-	RBM	RBMIn-	RBM	RBMIn-	RBM	RBMIn-
Acc	78.63	78.88	79.10	79.13	77.95	78.09	78.60	78.85
CB red	11.28	0.17	16.26	6.68	55.82	50.99	36.66	29.75

Table 3: *RBMining* and *RBM* average performance on 30 datasets

Table 3 illustrates the effect of adding the case base mining learning technique to the preprocessing RBM methods. In all the setup configurations, *RBMining* increases the accuracy of the classification. At the price of an increased computational cost, *RBMining* improves the accuracy of a model built on Reputation compared to its preprocessing methods.

4.4.2. RBM_ACBR performance

According to the test results in Table 2, *RBM_ACBR* achieved the highest average accuracy in comparison to the plain CBR and the proposed methods.

²The case base reduction for preprocessing methods is from 100% of the training data and, for learning methods, it is from 100% of the training + 10% of the testing data.

600 The small computational cost for the ACBR method had a very positive impact on the RBM model. *RBM_ACBR* also outperformed the state-of-the-art methods on average accuracy. The best accuracy among the state-of-the-art methods in the tests was obtained by the RDCL method with an average of 78.46%. *RBM_ACBR* achieved an accuracy of 79.31% while removing on average
605 3.57% more cases from the case base.

The ACBR configuration used in the combined method was the Minimal Goodness (MG) maintenance policy with oblivion to remove the incompetent cases and it was applied on a neighborhood size of 3 for its goodness calculations. By comparing the preprocessing method of *RBM_{nr}* with the combinational
610 method, we see that in 20 out of 30 data sets in the tests, adding the *ACBR* to the *RBM* led to an equal or higher accuracy than the CBR. This indicates the synergy between these two methods and how they can complement each other. Moreover, The *RBM_ACBR* method shows high confidence in maintenance of a CBR in both preprocessing stage and learning. We consider that this method
615 still can see improvements in its case base reduction to be more scalable on long runs and large data sets and our future work will continue in this direction.

4.5. Statistical analysis

The evaluation of the RBM model shows its positive effects on the average accuracy of a CBR technique. To measure the significance of these changes, the
620 best configuration of each proposed methods was put into a statistical analysis along with a plain CBR and the state-of-the-art methods. The Friedman [35] and Nemenyi [36] tests were selected to calculate the significance in the results. The Friedman test is a non-parametric statistical test of multiple comparisons that ranked the methods in the evaluation. In the Friedman test, the lower mean
625 rank translates to a better performance of an algorithm. The Nemenyi post-hoc test is used to detect the significance in the Friedman’s test results when all the classifiers are compared to each other. Figure 5 details the outcome of the Friedman and Nemenyi tests with 95% confidence in the methods within the scope of this paper. The blue lines in the figure show the significant threshold
630 of their corresponding method.

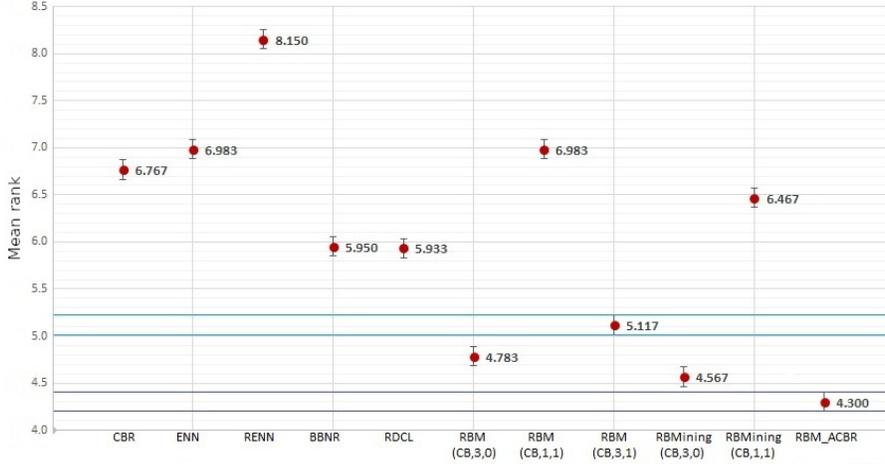


Figure 5: Friedman and Nemenyi statistical analysis results

According to Figure 5, all of the proposed methods based on RBM_{nr} ($RBM(CB, k, 0)$) with the purpose of noise removal achieved significant improvements in accuracy not only compared to the baseline CBR but also to all the other state-of-the-art methods implemented in the evaluation. The large gap between the mean ranks of the $RBM(CB, 3, 0)$, $RBM(CB, 3, 1)$, $RBMining(CB, 3, 0)$, and RBM_ACBR and the other state-of-the-art methods demonstrates the higher consistency of the improvements made by these methods. In the evaluation of RBM_{cr} -based methods it should be taken into consideration that the goal of these methods is to maintain the accuracy and reduce to the maximum the size of the case base. Two variations of it, $RBM(CB, 1, 1)$ and $RBMining(CB, 1, 1)$, fulfilled their purpose by achieving a similar score the baseline model in the Friedman and Nemenyi accuracy tests. On the other hand, $RBM(CB, 3, 1)$ is in the leading group in the statistical analysis test, along with the RBM_{nr} based methods, and achieved as much better score than the state-of-the-art methods. The best performance, however, belongs to the combinational method of RBM_ACBR whose evaluation shows that it is significantly better than any other methods in the test, including the baseline CBR, the state-of-the-art algorithms, and the other proposed methods.

5. Conclusion

In this paper, the RBM case base maintenance model is introduced. Several CBM methods based on the RBM model were proposed and their performance was tested. The evaluation of the proposals demonstrates the competence of the RBM in both the competence enhancement and competence preservation categories. RBM_{nr} achieves significantly better accuracy than the baseline CBR and the implemented state-of-the-art methods. RBM_{cr} with the main purpose of case base reduction also improves the average accuracy of the CBR technique

while significantly reducing the size of the case base. A variation of the RBM_{cr} (RBM(CB,3,1)) outperforms the state-of-the-art methods in terms of accuracy and removes around 36% of the case base. $RBMining$ methods add a learning
660 strategy to the RBM preprocessing methods. The case base mining results show that its algorithm allows the RBM to improve its case base over exposures to new cases. RBM_{nr} and RBM_{cr} based $RBMining$, maintained higher accuracy and a better ranking in the statistical analysis than their preprocessing methods. Finally, the RBM_ACBR method, which obtained the highest average accuracy
665 also obtained the best results in the statistical analysis. The $ACBR$ method is a good addition to the preprocessing method of the RBM_{nr} and the combined method achieved a significantly better average accuracy than the other methods implemented for this paper.

In conclusion, the reputation model is an effective maintenance method with
670 a high capability in the different types of maintenance strategies applied in a CBR. The simplicity of the implementation of the RBM methods is another positive point for the reputation model. We have also demonstrated that the performance of the preprocessing models of RBM improve with learning strategies.

The RBM model in its different variations significantly outperforms the plain
675 CBR, and the state-of-the-art methods implemented in this paper. However, we are considering several directions for future work. First, we would like to point out that the investigation on the datasets that RBM model did not improve the baseline accuracy. We will also experiment further with RBM on
680 unbalanced datasets. Finally, the $RBMining$ method can also benefit from additional research, since we consider it to be a potent algorithm that can be largely optimized in its Memory bank structure and maintenance.

References

- [1] M. M. Richter, R. O. Weber, Case-Based Reasoning: A Textbook, Springer, 2013
685 (2013).
- [2] A. Aamodt, E. Plaza, Case-Based Reasoning: Foundations Issues, Methodological Variations, and System Approaches, in: AI Communications, Vol. 7, 1994, pp. 39–59 (1994).
- [3] D. B. Leake, D. C. Wilson, Categorizing case-base maintenance: Dimensions
690 and directions, in: European Workshop on Advances in Case-Based Reasoning, Springer, 1998, pp. 196–207 (1998).
- [4] D. C. Wilson, D. B. Leake, Maintaining case-based reasoners: Dimensions and directions, Computational Intelligence 17 (2) (2001) 196–213 (2001).
- [5] H. Brighton, C. Mellish, Advances in instance selection for instance-based learning
695 algorithms, Data mining and knowledge discovery 6 (2) (2002) 153–172 (2002).
- [6] D. R. Wilson, T. R. Martinez, Reduction techniques for instance-based learning algorithms, Machine learning 38 (3) (2000) 257–286 (2000).
- [7] D. Dheeru, E. Karra Taniskidou, UCI machine learning repository (2017).
URL <http://archive.ics.uci.edu/ml>
- 700 [8] P. Hart, The condensed nearest neighbor rule (corresp.), IEEE transactions on information theory 14 (3) (1968) 515–516 (1968).

- [9] D. L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, *IEEE Transactions on Systems, Man, and Cybernetics* (3) (1972) 408–421 (1972).
- [10] G. Ritter, H. Woodruff, S. Lowry, T. Isenhour, An algorithm for a selective nearest neighbor decision rule (corresp.), *IEEE Transactions on Information Theory* 21 (6) (1975) 665–669 (1975).
- [11] G. Gates, The reduced nearest neighbor rule (corresp.), *IEEE transactions on information theory* 18 (3) (1972) 431–433 (1972).
- [12] D. W. Aha, D. Kibler, M. K. Albert, Instance-based learning algorithms, *Machine learning* 6 (1) (1991) 37–66 (1991).
- [13] C.-H. Chou, B.-H. Kuo, F. Chang, The generalized condensed nearest neighbor rule as a data reduction method, in: *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, Vol. 2, IEEE, 2006*, pp. 556–559 (2006).
- [14] I. Tomek, An experiment with the edited nearest-neighbor rule, *IEEE Transactions on systems, Man, and Cybernetics* (6) (1976) 448–452 (1976).
- [15] J. S. Sánchez, R. Barandela, A. I. Marqués, R. Alejo, J. Badenas, Analysis of new techniques to obtain quality training sets, *Pattern Recognition Letters* 24 (7) (2003) 1015–1022 (2003).
- [16] N. Segata, E. Blanzieri, S. J. Delany, P. Cunningham, Noise reduction for instance-based learning with a local maximal margin approach, *Journal of Intelligent Information Systems* 35 (2) (2010) 301–331 (2010).
- [17] E. Blanzieri, F. Melgani, An adaptive svm nearest neighbor classifier for remotely sensed imagery, in: *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on, IEEE, 2006*, pp. 3931–3934 (2006).
- [18] E. Blanzieri, F. Melgani, Nearest neighbor classification of remote sensing images with the maximal margin principle, *IEEE Transactions on geoscience and remote sensing* 46 (6) (2008) 1804–1811 (2008).
- [19] N. Segata, E. Blanzieri, P. Cunningham, A scalable noise reduction technique for large case-based systems, in: *International Conference on Case-Based Reasoning, Springer, 2009*, pp. 328–342 (2009).
- [20] A. Smiti, Z. Elouedi, Wcoid: Maintaining case-based reasoning systems using weighting, clustering, outliers and internal cases detection, in: *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on, IEEE, 2011*, pp. 356–361 (2011).
- [21] P. Bajcsy, N. Ahuja, Location-and density-based hierarchical clustering using similarity analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (9) (1998) 1011–1015 (1998).
- [22] A. Smiti, Z. Elouedi, Coid: Maintaining case method based on clustering, outliers and internal detection, in: *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2010, Springer, 2010*, pp. 39–52 (2010).
- [23] B. Smyth, M. T. Keane, Remembering to forget, in: *Proceedings of the 14th international joint conference on Artificial intelligence, Citeseer, 1995*, pp. 377–382 (1995).
- [24] E. McKenna, B. Smyth, Competence-guided editing methods for lazy learning, in: *Proceedings of the 14th European conference on artificial intelligence, IOS Press, 2000*, pp. 60–64 (2000).
- [25] H. Brighton, C. Mellish, Advances in instance selection for instance-based learning algorithms, *Data mining and knowledge discovery* 6 (2) (2002) 153–172 (2002).
- [26] S. J. Delany, P. Cunningham, An analysis of case-base editing in a spam filtering system, in: *European Conference on Case-Based Reasoning, Springer, 2004*, pp. 128–141 (2004).

- [27] S. J. Delany, The good, the bad and the incorrectly classified: Profiling cases for case-base editing, in: International Conference on Case-Based Reasoning, Springer, 2009, pp. 135–149 (2009).
- 755 [28] S. J. Delany, N. Segata, B. Mac Namee, Profiling instances in noise reduction, Knowledge-Based Systems 31 (2012) 28–40 (2012).
- [29] M.-K. Haouchine, B. Chebel-Morello, N. Zerhouni, Competence-preserving case-deletion strategy for case-base maintenance., in: ECCBR’08, Vol. 1, 2008, pp. 171–184 (2008).
- 760 [30] B. Chebel-Morello, M. K. Haouchine, N. Zerhouni, Case-based maintenance: Structuring and incrementing the case base, Knowledge-Based Systems 88 (2015) 165–183 (2015).
- [31] N. Lu, G. Zhang, J. Lu, Concept drift detection via competence models, Artificial Intelligence 209 (2014) 11–28 (2014).
- 765 [32] M. Salamó, M. López-Sánchez, Adaptive case-based reasoning using retention and forgetting strategies, Knowledge-Based Systems 24 (2) (2011) 230–247 (2011).
- [33] M. Sánchez-Marrè, U. Cortés, J. Béjar, I. R. Roda, M. Poch, Reflective reasoning in a case-based reasoning agent, in: Collaboration between Human and Artificial Societies, Springer, 1999, pp. 142–158 (1999).
- 770 [34] R. Pan, Q. Yang, S. J. Pan, Mining competent case bases for case-based reasoning, Artificial Intelligence.
- [35] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the american statistical association 32 (200) (1937) 675–701 (1937).
- 775 [36] P. Nemenyi, Distribution-free multiple comparisons, in: Biometrics, Vol. 18, International Biometric Soc 1441 I ST, NW, Washington DC 20005-2210, 1962, p. 263 (1962).