# A Hybrid Method for Fast Finding the Reduct with the Best Classification Accuracy

Mehmet HACIBEYOGLU[1], Ahmet ARSLAN[2], Sirzat KAHRAMANLI[3]
[1]*Department of Computer Engineering, Necmettin Erbakan University, Konya, Turkey*
[2]*Department of Computer Engineering, Selcuk University, Konya, Turkey*
[3]*Department of Computer Engineering, Mevlana University, Konya, Turkey*
*hacibeyoglu@konya.edu.tr*

***Abstract*—Usually a dataset has a lot of reducts finding all of which is known to be an NP hard problem. On the other hand, different reducts of a dataset may provide different classification accuracies. Usually, for every dataset, there is only a reduct with the best classification accuracy to obtain this best one, firstly we obtain the group of attributes that are dominant for the given dataset by using the decision tree algorithm. Secondly we complete this group up to reducts by using discernibility function techniques. Finally, we select only one reduct with the best classification accuracy by using data mining classification algorithms. The experimental results for datasets indicate that the classification accuracy is improved by removing the irrelevant features and using the simplified attribute set which is derived from proposed method.**

***Index Terms*—artificial intelligence, classification algorithms, decision trees, discernibility function, feature selection**

## I. INTRODUCTION

Usually the rules for classification of a dataset are generated starting only from a set of seen examples (objects, instances) named as a training set. This approach to the rule generation is called as rule induction [1,2] the objective of which is to learn from the training set how to classify the new (unseen) examples [3]. In the other words, the main task of the rule induction is to generate from the training set such a set of rules that to be also able to classify the new examples [4,5]. There are many approaches to rule induction the most common of which are Sequential Covering-based, Decision Tree-based and Rough Set-based ones.

The sequential covering is the basic strategy of the iterative algorithms CN2, AQ [6,7] and RIPPER [8]. An algorithm of this family forms a rule in each of iterations and removes from the training set all the positive examples covered by this rule. It is iterated until all examples are covered. Despite successes of these algorithms in some applications, they are known to be imperfect [4,9,10] due to that they are insensitive for the conditions describing small parts of the examples and do not perceive noise [11,12].

A decision tree is a hierarchically organized set of nodes each of which is associated with a test on the values of an attribute. In a decision tree, each edge from a non-leaf node is labeled with a particular value of the attribute and each leaf node is associated with a certain class. In order to reduce the tree as more as possible and to increase its ability of classification of new examples, every non leaf node with the information gain less than the predefined threshold value is replaced with the leaf nodes that under it. This allows classifying every new example into one of the predefined classes [13].

The major advantage of a decision tree-based algorithm (ID3, C4.5, C5.0 and CART) [14,15] is that its each path represents a single IF-THEN rule. Unfortunately, the decision tree-based algorithms do not perceive noise and contradictions in data because they assume that Bayes error rate for the classification problem is zero. Additionally, their classification ability is seriously affected by the pruning that cannot be always done perfectly [11,12,16].

The Rough Set-based rule induction is founded on the notion of reduct that is a minimal subset of attributes with the same classification power as the original set of attributes with respect to the given decision function. Once a selected reduct (SR) has been computed, the rules are easily constructed by overlaying it over the originating training set and reading off the values [17,18]. That is, the Rough Set-based rule induction problem is solved mainly via feature selection problem [19,20,21]. Since a dataset may have different reducts of different sizes, a SR can be recognized only in comparison to other ones. But since the finding all reducts (AR) of a dataset is known to be NP-hard, most of the feature selection algorithms generate only one of reducts with risk of overlooking the optimal ones [17,22,23]. Therefore, usually the classification accuracy level of the reduct obtained by such a way is very low [10,20].

Notice that all of above-mentioned rule induction approaches have a common drawback such that when in a training set the class distribution is highly skewed, the majority class can be adequately represented, but the minority class may be neglected. As a result, they usually have poor classification accuracy for the examples from the minority class. However, highly imbalanced problems generally have highly non uniform error costs that often favor the minority class of primary interest [20]. Our studies show that the problem of finding the reduct with the best classification accuracy (RWBCA) can be solved by a decision tree and rough set- based hybrid model working as follows:

1. The attributes dominant in a given dataset is obtained by using the decision tree techniques,
2. The reducts with the obtained dominant attributes are generated by using the discernibility function techniques widely used in rough set theory and applications,
3. The RWBCA is selected by testing the obtained reducts to the classification accuracy.

Digital Object Identifier 10.4316/AECE.2013.04010

In sequel, instead of the concept training set, we will use the more comprehensive concept dataset.

The rest of paper is organized as follows. In Chapter II, the relation between the group of dominant attributes and the RWBCA is explained. In Chapter III, the method generating a group of dominant attributes is given. In Chapter IV the generation of the group of dominant attributes-based reducts is explained. In Chapter V, the results of the experiments over the datasets from the UCI repository are shown. The paper is concluded with a summary in Chapter VI.

## II. The Relation between the Group of Dominant Attributes and the Reduct with the Best Classification Accuracy Level

Usually the size of a reduct is significantly less than the size of the set A of all attributes describing the dataset N. In general, if a dataset have many attributes then the classification rules of this dataset have many rules. Therefore, it is convenient first to obtain a shortest reduct for the given dataset and then generate the classification rules based on it [24,25]. In [26], it is stated that the rules generated by reducts are often too specific and cannot classify new objects. Moreover, our experiments with a lot of datasets show that even different reducts of the same size may provide different classification accuracy and the RWBCA is not always a shortest reduct. Therefore, for obtaining the RWBCA with respect to a given dataset it is necessary to: 1) generate ARs of this dataset, 2) obtain the classification accuracy provided by each of them and 3) select the RWBCA. But unfortunately, this is a very complex problem, because the number of ARs of a dataset may be as large as $\binom{n}{n/2} = O(2^n)$ [27,28]. On the other hand any dataset has a reduct H that provides the better classification accuracy than other ones. In order to obtain the H, firstly we obtain the group of attributes that must be contained in H with a great probability. For this aim, we use the heuristic widely employed in such the popular data classification algorithms as ID3 [29], C4.5 [30] and CN2 [31]. We call the group of the attributes obtained by proposed method as group of dominant attributes (*GDA*). Unfortunately, the result of processing a lot of dataset show that the *GDA* obtained for a dataset usually is insufficient for classification of the dataset with acceptable accuracy. The reason of this is that the *GDA* generated for a certain dataset describes the only most powerful path of the decision tree needed to classify this dataset. However, the *GDA* obtained for a dataset allows us to generate not ARs for a dataset, but only those of them that contain this GDA. We call every reduct containing the *GDA* as a *GDA-based reduct (GDAR)*. The maximal possible number of *GDAR* can be obtained as follows:

$$GDAR = \sum_{i=0}^{\alpha-\beta} \binom{\alpha-\beta}{i} = 2^{\alpha-\beta} \tag{1}$$

where, $\beta$ and $\alpha$ are the sizes of the GDA and the reduct of maximal size containing this GDA, respectively. It is obvious that always $\beta \leq \alpha$. This approach reduces the number of *GDAR* from $2^n$ to $2^n/2^{\alpha-\beta}=2^{n-(\alpha-\beta)}$, where $2^n$ is the maximal possible number of ARs of a dataset with *n* attributes [23,34]. Since n, $\alpha$ and $\beta$ may take diverse values for different datasets, we can estimate the values of *R* as:

$$R = \frac{Number \text{ of ARs}}{Number \text{ of GDARs}} \tag{2}$$

For example, the values of R for some datasets from the UCI repository are given in Table 1.

TABLE 1. The values of *R* for some datasets

| Dataset | Selected GDA | Number of | | R |
|---|---|---|---|---|
| | | ARs | GDARs | |
| Lymphography | a1a2a6a13a14 | 424 | 14 | **30.3** |
| Zoo | a9a12a13 | 34 | 4 | **8.5** |
| Breast Cancer Wisconsin | a1a2a6 | 19 | 5 | **3.8** |
| Tictoctoe | a3a5a7 | 8 | 5 | **1.6** |
| Chess | a1a6a10a15a21 a32a33a35 | 4 | 2 | **2** |
| Voting | a3a4a11 | 3 | 2 | **1.5** |
| Monks | a1a2a5 | 1 | 1 | **1** |
| Solar Flare | a1a2a6 | 1 | 1 | **1** |
| Primary Tumor | a2a4a5a7a9a14 | 1 | 1 | **1** |

## III. Generating the Group of Dominant Attributes

Usually the dataset N is given by a table the rows and columns are labeled by the examples and attributes, respectively. The effect of an attribute $A_i \in A$ to the classification accuracy of the dataset to which it belongs may be determined via the information gain of this attribute [1]. The information gain of an attribute is reducing effect of this attribute on the entropy of the dataset to which it belongs. The entropy *E(N)* of a dataset *N* and the information gain $I(A_i)$ of an attribute $A_i \in A$ can be calculated by the following formulas, respectively [35].

$$E(N) = \sum_{k=1}^{g} - P_k \log_2 P_k \tag{3}$$

$$I(A) = E(N) - \sum_{v \in Values(A)} \frac{|N(v)|}{|N|} E(N(v)) \tag{4}$$

where, $P_k$ is the proportion of examples labeled by the class $C_{k \in \{1,2,\ldots,g\}}$ and *N(v)* is the subset of examples in *N* for which $A_i = v$. For ease of computations by the formula (4), it can be partitioned into the following tree formulas:

$$E(v_i^j) = \frac{n_i^j}{n_t} \times \sum_{k=1}^{G} - \frac{n_i^j(C_k)}{n_j^i} \times \log_2 \frac{n_i^j(C_k)}{n_i^j} \tag{5}$$

$$E(A_i) = \sum_{k=1}^{V_i} E(v_i^j) \tag{6}$$

$$I(A_i) = E(N) - E(v_i^j\}) \tag{7}$$

where, $v_i^j$ is the *j*-th value of the attribute $A_i$, $E(v_i^j)$ is the entropy for the value $v_i^j$, $n_i^j$ is the number of examples with $A_i = v_i^j$, $n_t$ is the total number of examples in *N*, *G* is the number of classes in *N* and $n_i^j(C_k)$ is the number of examples with $A_i = v_i^j$ and class-label $C_k$. The formulas (5), (6) and (7) will be used for obtaining the attribute with the highest information gain.

As it is well known in decision tree, every selected attribute S splits the dataset N into the parts $T_1, T_2, \ldots, T_{|V(S)|}$. Hence, in order to obtain the next element of the GDA first we obtain the attribute $S \in A$ with the highest information gain Ij(S) for each $T_j \in \{1,2, \ldots, |V(S)|\}$. Then we select the sub dataset $T \in \{T_1, T_2, \ldots, T_{|V(S)|}\}$ with $IG_{max}=max\{I_j(S)\}$ as the new state of N to be processed in the next iteration. This process is repeated until there a sub dataset T with attributes of zero information gains occurs. The algorithm *Generate_GDA* implementing this process is given in Figure 1.

```
Generate_GDA(N,A)
1. Calculate the entropy of the original
   attribute set of dataset N
2. Calculate the information gains of
   attributes in dataset N and select the
   attribute with the highest information
   gain
3. GDA=∅ ; IGmax=0
4. Do
   {
     4.1. GDA=GDA∪S; A=A-S; Vs = (v_s^j)_{j=1}^{|V(S)|}
     4.2. For j=1 to | Vs |
     {
       4.2.1. Tj = ∀ E∈N : E(S) =v_s^j ; N=Tj
       4.2.2. Calculate the information
              gains of attributes in
              subset Tj and select the
              attribute with the highest
              information gain
       4.2.3. If Ij(S) > IGmax then IGmax =
              Ij(S); T=Tj
     }
     4.3. N=T
   }
   while IGmax > 0
5. RETURN (GDA)
```

Figure 1. The algorithm generating the GDA for a dataset

In the algorithm *Generate _GDA (Figure 1)*, the statement 1 uses the formula (3) that calculates the entropy of the original attribute set (OAS) *N*. The statement 2 select the attribute $S \in A$ with the highest information gain *I(S)* in the OAS *N*. The statement 3 assigns initial values to the GDA and $IG_{max}$. The statement 4.1 of the *Do-While loop* includes the selected attribute *S* into the GDA, removes it from the attribute set *A* and inputs the set $V_s$ of the values of the attribute *S*. The statements from 4.2.1 to 4.2.3 form the body of the *For loop* in which the statement 4.2.1 forms a subset $T_j \subset N$ containing all examples with $E(S)= v_s^j$. The statement 4.2.2 calculates the information gains of the attributes and select the attribute *S* with the highest information gain $I_j(S)$ in the dataset $T_j$. The statement 4.2.3 compares the values of the last obtained $I_j(S)$ and $IG_{max}$. If $I_j(S)>IG_{max}$ then the assignments $IG_{max}=I_j(S)$ and $T=T_j$ are made. The statement 3.3 assigns *T* to *N* as the new state of the dataset *N* to be processed in the next iteration of the *Do-While loop*. Consequently, this procedure gives us the decision tree path whose total information gain is the maximum and includes the dominant attributes of the dataset *N*.

**Example 1.** Obtain the GDA for the dataset given in Table 2 (adopted from [32])

TABLE 2. AN EXAMPLE OF A DATASET

| Examples | Condition Attributes | | | | | Decision | | |
|---|---|---|---|---|---|---|---|---|
| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $D_1$ | $D_2$ | $D_3$ |
| $N_1$ | 1 | 2 | 1 | 2 | 3 | 1 | 0 | 0 |
| $N_2$ | 1 | 3 | 0 | 4 | 6 | 1 | 0 | 0 |
| $N_3$ | 1 | 3 | 2 | 2 | 8 | 0 | 1 | 0 |
| $N_4$ | 0 | 2 | 1 | 2 | 3 | 0 | 0 | 1 |
| $N_5$ | 1 | 3 | 1 | 4 | 6 | 0 | 1 | 0 |
| $N_6$ | 1 | 2 | 7 | 8 | 9 | 1 | 0 | 0 |
| $N_7$ | 1 | 2 | 0 | 2 | 3 | 0 | 0 | 1 |

According to *Generate_GDA* the *GDA* for the dataset $N=\{N_1,\ldots,N_7\}$ given in Table 2 can be generated as follows:

**The statement 1**: Calculating the entropy of the OAS *N*. *E(N)= 1,557*.

**The statements 2:** The information gains for the attributes $A_1$, $A_2$, $A_3$, $A_4$ and $A_5$ calculated: $I(A_1)= 0{,}306$, $I(A_2)=0.592$, $I(A_3)=0.592$, $I(A_4)=0.414$ and $I(A_5)=0.877$. Selecting the attribute with the highest information gain. Since $max\{I(A_1), I(A_2), I(A_3), I(A_4), I(A_5)\}=I(A_5)$, as the dominant attribute $S=A_5$ is to be chosen.

**The first iteration** of the *Do-While loop* of the algorithm *Generate_GDA*.

**The statement 4.1:** Adding the selected attribute $S=A_5$ to the set GDA, removing the attribute $A_5$ from the set *A* and declaring the set $V_5$ of values of the attribute $A_5$.

$GDA= \{A_5\}$; $A = \{A_1, A_2, A_3, A_4, A_5\} - A_5 = \{A_1, A_2, A_3, A_4\}$;
$V_5 = \{v_5^1, v_5^2, v_5^3, v_5^4\} = \{3,6,8,9\}$

**The statement 4.2:** Selecting the sub dataset containing the attribute with the highest information gain.

**The first iteration** of the *For loop* of the algorithm *Generate_GDA*.

**The sub statement 4.2.1:** Obtaining the sub dataset $T_1 \subset N$ with $v_5^1 = 3$.

TABLE 3.1 THE SUB DATASET $T_1$ CONTAINING THE EXAMPLES WITH $A_5$=3

| Examples | Condition Attributes | | | | | Decision | | |
|---|---|---|---|---|---|---|---|---|
| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5^*$ | $D_1$ | $D_2$ | $D_3$ |
| $N_1$ | 1 | 2 | 1 | 2 | $3^*$ | 1 | 0 | 0 |
| $N_4$ | 0 | 2 | 1 | 2 | $3^*$ | 0 | 0 | 1 |
| $N_7$ | 1 | 2 | 0 | 2 | $3^*$ | 0 | 0 | 1 |

**The sub statement 4.2.2:** Obtaining the attribute with the highest information gain for the subset $T_1$. The sub statement 4.2.2 generates information gains for all attributes for the subset $T_1$ and selects the attribute $S \in A$ with the highest information gain $I_1(S)$: $I_1(A_1)=0.25$, $I_1(A_2)=0.92$,

$I_1(A_3)=0.25$ and $I_1(A_4)=0.92$. As the highest from these values either $I_1(A_2)=0.92$ or $I_1(A_4)=0.92$ may be selected. Suppose that the algorithm chose $I_1(A_2)=0.92$ and $S=A_2$.

**The sub statement 4.2.3:** Updating $IG_{max}$ and $T$. Since $I_1(A_2)=0.92>IG_{max}=0$, $IG_{max}=I_1(A_2)=0.92$ and $T=T_1$.

**The second iteration** of the For loop of the algorithm Generate_GDA.

**The sub statement 4.2.1:** Obtaining the sub dataset $T_2 \subset N$ with $v_5^2 = 6$.

TABLE 3.2. THE SUB DATASET $T_2$ CONTAINING THE EXAMPLES WITH $A_5$=6

| Examples | Condition Attributes | | | | | Decision | | |
|---|---|---|---|---|---|---|---|---|
| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5^*$ | $D_1$ | $D_2$ | $D_3$ |
| $N_2$ | 1 | 3 | 0 | 4 | $6^*$ | 1 | 0 | 0 |
| $N_5$ | 1 | 3 | 1 | 4 | $6^*$ | 0 | 1 | 0 |

**The sub statement 4.2.2:** Obtaining the attribute $S \in A$ with the highest information gain $I_2(S)$ for the sub dataset $T_2$. This is done by the same way as in the same statement of the first iteration of the *For loop*: $I_2(A_1)=0$, $I_2(A_2)=0$, $I_2(A_3)=1$ and $I_2(A_4)=0$. Since $max(I_2(A_1), I_2(A_2), I_2(A_3), I_2(A_4))=I_2(A_3)=1$, there the dominant attribute is to be $S=A_3$.

**The sub statement 4.2.3:** Updating $IG_{max}$ and $T$. Since $I_2(A_3)=1>IG_{max}=0.92$, $IG_{max}=I_2(A_3)=1$ and $T=T_2$.

**The third iteration** of the *For loop* of the algorithm *Generate_GDA.*

**The sub statement 4.2.1:** Obtaining the sub dataset $T_3 \subset N$ with $v_5^3 = 8$.

TABLE 3.3. THE SUB DATASET $T_3$ CONTAINING THE EXAMPLES WITH $A_5$=8

| Examples | Condition Attributes | | | | | Decision | | |
|---|---|---|---|---|---|---|---|---|
| | $A_1$ | $A_2$ | $A_3^*$ | $A_4$ | $A_5^*$ | $D_1$ | $D_2$ | $D_3$ |
| $N_3$ | 1 | 3 | 2 | 2 | $8^*$ | 0 | 1 | 0 |

**The sub statement4.2.2:** Obtaining the attribute $S \in A$ with the highest information gain $I_3(S)$ for the sub dataset $T_3$. This is done by the same way as in the same statement of the previous iterations of the *For loop*: $I_3(A_1)=0$, $I_3(A_2)=0$; $I_3(A_3)=0$ and $I_3(A_4)=0$. Since all of these values are equivalent, any of them may be selected as highest one. Suppose that the algorithm chose $I_3(A_1)=0$ and $S=A_1$.

**The sub statement 4.2.3:** Updating $IG_{max}$ and $T$. Since $I_3(A_1)=0<IG_{max}=1$, $IG_{max}$ and $T$ remain unchanged.

**The fourth iteration** of the *For loop* of the algorithm *Generate_GDA.*

**The sub statement 4.2.1:** Obtaining the sub dataset $T_4 \subset N$ with $v_5^4 = 9$.

TABLE 3.4. THE SUB DATASET $T_4$ CONTAINING THE EXAMPLES WITH $A_5$=9

| Examples | Condition Attributes | | | | | Decision | | |
|---|---|---|---|---|---|---|---|---|
| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5^*$ | $D_1$ | $D_2$ | $D_3$ |
| $N_6$ | 1 | 2 | 7 | 8 | $9^*$ | 1 | 0 | 0 |

**The sub statement 4.2.2:** Obtaining the attribute $S \in A$ with the highest information gain $I_4(S)$ for the sub dataset $T_4$. This is done by the same way as in the same statement of the previous iterations of the *For loop*: $I_4(A_1)=0$, $I_4(A_2)=0$, $I_4(A_3)=0$ and $I_4(A_4)=0$. Since all of these values are equivalent, any of them may be selected as highest one. Suppose that the algorithm chose $I_4(A_1)=0$ and $S=A_1$.

**The sub statement 4.2.3:** Updating $IG_{max}$ and $T$. Since $I_4(A_1)=0<IG_{max}=1$, the $IG_{max}$ and $T$ remain unchanged.

**The statement 4.3** of the algorithm *Generate_GDA*: Since $IG_{max}=I_2(A_3)=1$ is greater than *0*, as the dominant attribute $S=A_3$ is to be chosen and in the next iteration of the *Do-While loop* the dataset $N=T$ will be processed.

**The second iteration** of the *Do-While loop* of the algorithm *Generate_GDA.*

**The statement 4.1:** Adding the selected attribute $S=A_3$ to the set GDA, removing the attribute $A_3$ from the set $A$ and declaring the set $V_3$ of values of the attribute $A_3$.

$GDA=\{A_3, A_5\}$; $A=\{A_1, A_2, A_3, A_4\}-A_3=\{A_1, A_2, A_4\}$; $V_3 = \{v_3^1, v_3^2\} = \{0,1\}$

**The statement 4.2:** Selecting the sub dataset containing the attribute with the highest information gain.

**The first iteration** of the *For loop* of the algorithm *Generate_GDA.*

**The sub statement 4.2.1:** Obtaining the sub dataset $T_1 \subset N$ with $v_3^1 = 0$.

TABLE 4.1. THE SUB DATASET $T_1$ CONTAINING THE EXAMPLES WITH $A_3$=0

| Examples | Condition Attributes | | | | | Decision | | |
|---|---|---|---|---|---|---|---|---|
| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5^*$ | $D_1$ | $D_2$ | $D_3$ |
| $N_2$ | 1 | 3 | $0^*$ | 4 | $6^*$ | 1 | 0 | 0 |

**The sub statement 4.2.2:** Obtaining the attribute $S \in A$ with the highest information gain $I_1(S)$ for the sub dataset $T_1$. This is done by the same way as in the same statement of the previous iterations of the *For loop*: $I_1(A_1)=0$, $I_1(A_2)=0$ and $I_1(A_4)=0$. Since all of these values are equivalent, any of them may be selected as highest one. Suppose the algorithm chose $I_1(A_1)=0$ and $S=A_1$.

**The substatement 4.2.3:** Updating $IG_{max}$ and $T$. Since $I_1(A_1)=IG_{max}=0$, $IG_{max}$ and T remain unchanged.

**The second iteration** of the *For loop* of the algorithm *Generate_GDA(N,A).*

**The substatement 4.2.1:** Obtaining the sub dataset $T_2 \subset N$ with $v_3^2 = 1$.

TABLE 4.2. THE SUB DATASET $T_2$ CONTAINING THE EXAMPLES WITH $A_3$=1

| Examples | Condition Attributes | | | | | Decision | | |
|---|---|---|---|---|---|---|---|---|
| | $A_1$ | $A_2$ | $A_3^*$ | $A_4$ | $A_5^*$ | $D_1$ | $D_2$ | $D_3$ |
| $N_5$ | 1 | 3 | $1^*$ | 4 | $6^*$ | 0 | 1 | 0 |

**The substatement 4.2.2:** Obtaining the attribute $S \in A$ with the highest information gain $I_2(S)$ for the sub dataset $T_1$. This is done by the same way as in the same statement of the previous iterations of the *For loop*: $I_2(A_1)=0$, $I_2(A_2)=0$ and $I_2(A_4)=0$. Since all of these values are equivalent, any of them may be selected as highest one. Suppose that the algorithm chose $I_2(A_1)= 0$ and $S=A_1$.

**The substatement 4.2.3:** Updating $IG_{max}$ and $T$. Since $I_2(A_1)= IG_{max} = 0$, $IG_{max}$ and $T$ remain unchanged.

**The statement 4.3** of the algorithm *Generate_GDA*: Since $IG_{max} = I_2(A_1) = 0$, there is no attribute to be chosen as dominant one. That is, the *GDA* generating process has been completed with the result $GDA=\{A_5, A_3\}$.

As mentioned above, the *GDA* generated for a dataset is not sufficient for classification of this dataset with an acceptable accuracy. Therefore, in order to achieve classification accuracy as high as possible, the obtained *GDA* has to be completed with the other attributes not contained in it. In order to obtain which attributes to add to *GDA*, we generate ARs each of which contains the obtained *GDA*.

## IV. Generating the GDA-based reducts

Let us first remind the concepts of clause and discernibility function (DF) for a dataset.

**Definition 4.1.** A clause $H_{ik}$ is the difference between the examples $E_j \in N$ and $E_k \in N$ expressed as follows [33]:

$$H_{ik} = h_{jk1} \vee h_{jk2} \vee \ldots \vee h_{jk|N|} \tag{8}$$

where, $h_{jki} = A_i$ if $E_j(A_i) \neq E_k(A_i)$, $h_{jki} = 0$ if $E_j(A_i) = E_k(A_i)$ and $E_j(A_i)$, $E_k(A_i)$ are the values of the attribute $A_i$ in the examples $E_j$ and $E_k$, respectively.

A propositional clause given by the formula (8) can be represented by the following bit-based clause (BBC).

$$B_{ik} = b_{jk1} b_{jk2} \ldots b_{jki} \ldots b_{ik|N|} \tag{9}$$

where, $b_{jki} = 1$ if $E_j(A_i) \neq E_k(A_i)$ and $b_{jki} = 0$ otherwise.

**Definition 4.2.** The *Cartesian bitwise disjunction* of all possible BBCs derived from a certain dataset is called the *bit-based DF* for this dataset [33,34,36]. The *bit-based DF* for a dataset with $M$ objects may be expressed as follows:

$$DF = \mathop{\Big\vert}_{j=1}^{M-1} \left( \mathop{\Big\vert}_{k=j+1}^{M} B_{jk} \right) \tag{10}$$

where, $\vert$ is the sign of *bitwise disjunction (OR)* operation performed as $\{x \vert y / x \in B_X \text{ and } Y \in B_Y\}$. According to the formula (10), from a dataset with $M$ examples exactly $Z=0.5x(M^2-M)$ clauses can be derived. But there in a *DF* may be redundant clauses after removing which the *DF* of the size $Z$ can be reduced into the $DF_{min}$ of the size $Q \leq Z$ [17,33,37]. This event leads to disordering of the indexation used in the formulas (9) and (10). Therefore, it will be better to rewrite these formulas for $q \in \{1,2,\ldots,Q\}$ and $DF_{min}$ as follows:

$$B_q = b_{q1} b_{q2} \ldots b_{qi} \ldots b_{q|N|} \tag{11}$$

$$DF_{min} = \mathop{\Big\vert}_{q=1}^{Q} B_q \tag{12}$$

Note that every BBC of the type (11) is a *sum bit-vector* as well as every propositional clause of the type (8) is a *sum term*. For instance, the BBC representation of the clause $(A_5 \vee A_6 \vee A_8)$ is to be as 000011010. To associate a BBC in its propositional form, it may be used a *bit-field structure* of the following form:

***Struct_Clause*** *{Unsigned $A_1$:1; … ; Unsigned $A_N$:1}* (13)

In order to obtain *disjunctive normal form* (DNF) of $DF_{min}$ by the formula (12) it is necessary to expand each $B_q$, $q=1, 2, 3, \ldots, Q$ into the set $E(B_q)$ of its *unit clauses* [27] as follows:

$$E(B_q) = \left\{ \Pr_i B_q \mid b_{qi} = 1 \right\} \tag{14}$$

For example, if $B_q = 100110$ then $E(B_q)=\{Pr_i B_q / b_{qi}=1\} =\{100000, 000100, 000010\}$. Consequently, the formula (12) has to be rewritten as:

$$DF_{DNF} = \mathop{\Big\vert}_{q=1}^{Q} E(B_q) \tag{15}$$

Based on the expression (11), the BBCs to be processed by the formula (15) may be represented as a binary matrix given in Figure 2.

| | $A_1$ | $A_2$ | … | $A_N$ |
|---|---|---|---|---|
| $B_1$ | $b_{11}$ | $b_{12}$ | … | $b_{1N}$ |
| $B_2$ | $b_{21}$ | $b_{22}$ | … | $b_{2N}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $B_Q$ | $b_{Q1}$ | $b_{Q2}$ | … | $b_{QN}$ |

Figure 2. The structure of a DFM

In accordance to [38], we will call a binary matrix of type given in Figure 2 as a *discernibility function matrix (DFM)*. Each row of the DFM given in Figure 2 is a certain BBC $B_{q \in \{1,2,\ldots,Q\}}$ from the formula (12). Due to commutativity of the OR operation, the order of rows (BBCs) in a DFM is not important as well as the order of BBCs in the formula (12). Since we are interested in only those implicants that contain all attributes from the obtained *GDA*, we may reduce the BBCs containing these attributes by removing other attributes from them. The detailed information about the DFM of a $DF_{min}$ and deriving from it the reducts is given in [38]. In brief, such a reduction of a *DFM* is based on the parallel decomposition of Boolean functions according to which, for example, the function $DF = (a \vee b \vee c \vee d \vee e)F$ may be rewritten as follows:

$$DF = aF \vee bF \vee cF \vee dF \vee eF \tag{16}$$

where, *F* is a conjunction of some clauses containing different combinations of the variables *a, b, c, d* and *e.* Assume that, we have obtained the *GDA* consisting of the attributes *a* and *c*. This is to say that we are interested only in the first and third components of the expression (16). In this way, we can obtain all implicants containing the attributes *a* and *c* by the following reduced form of the expression (16).

$$DF_{red} = aF \vee cF \tag{17}$$

As it is seen from this example, *GDARs*, may be generated by the algorithm *Generate_GDARs* given below.

```
Generate_GDARs (DFM, A, GDA)
1. For i=1 to /GDA/
   {
     1.1. Select the i-th element of the
          GDA and fix the attribute S
          represented by this element
     1.2. Fix all rows of DFM in which
          S=1
     1.3. Preserve only one of these rows
     1.4. Reduce the preserved row by
          removing from it all 1's not
          associated with the attribute S
     1.5. Remove all rows from the DFM
          that absorbed by the reduced
          row
   }
2. Expand each row (BBC) of the reduced
   DFM by the formula (14)
3. Generate the bit-based DNF of the DF_DNF
   by the formula (15)
4. Convert the obtained DNF to the
   reducts in accordance with the bit-
   field association structure (13)
```

Figure 3. The algorithm generating the GDARs

Note that a row $R_j$ of a DFM is absorbed by another row $R_k$ of the same DFM if $R_j \,\&\, R_k = R_k$.

**Example 2.** In the Example 1, we obtained that the GDA for the dataset given in Table 2 contains the attributes $A_3$ and $A_5$. Let us to obtain the *reducts* containing this GDA. The associated bit-field structure using for this example is as follows:

***Struct_Clause*** *{Unsigned $A_1$:1; Unsigned $A_2$:1; Unsigned $A_3$:1; Unsigned $A_4$:1; Unsigned $A_5$:1}* (18)

The DF for this dataset is: DF = {*01111**, *01101**, 10000, 01011, *00111**, 00100, *00111**, *11111**, *00100**, *01111**, *01011**, *11101**, *00111**, *01111**, *01101**, *11011**, *10111**, *10100**, *01111**, *01111**, *00111**} = {10000, 01011, 00100}

where, the components marked by "*" are those absorbed by other ones removed from the result. Thus,

$$DF_{min} = \{10000, 01011, 00100\} \tag{19}$$

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ |
|---|---|---|---|---|---|
| $B_1$ | 1 | 0 | 0 | 0 | 0 |
| $B_2$ | 0 | 1 | 0 | 1 | 1 |
| $B_3$ | 0 | 0 | 1 | 0 | 0 |

Figure 4. The DFM representation of the $DF_{min}$ given by the expression (19)

Since GDA = {$A_3$ , $A_5$ }, we are interested with the *reducts* containing the attributes $A_3$ and $A_5$. According to the algorithm *Generate_GDARs* (Figure 3), first we have to fix all rows of the *DFM* (Figure 4) containing the attribute $A_3$, preserve only one of these rows and remove from it all 1's associated with attributes other than $A_3$. Since in the *DFM* exist only one row $B_3$ with a single 1 associated with $A_3$, the *DFM* remains unchanged with respect to this attribute. The processing of the *DFM* with respect to $A_5$ is the same as that with respect to the attribute $A_3$. In the *DFM*, the attribute $A_5$ is present only in the row $B_2$ that contains two more 1's associated with $A_2$ and $A_4$. After removing these 1's from the row $B_2$, we get the following reduced *DFM* (Figure 5).

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ |
|---|---|---|---|---|---|
| $B_1$ | 1 | 0 | 0 | 0 | 0 |
| $B_2$ | 0 | 0 | 0 | 0 | 1 |
| $B_3$ | 0 | 0 | 1 | 0 | 0 |

Figure 5. The DFM with rows from which the 1's associated with attributes other than $A_3$ and $A_5$ have been removed

By applying the formula (15) to the *DFM* in Figure 5 we get the following result:

$DF_{min}$=(10000) ¦ (00001) ¦ (00100) = 10101

According to association bit-field structure (18), the result bit-string $DF_{min}$=*10101* is to be interpreted as the reduct *{$A_1$, $A_3$, $A_5$}*. Note that the dataset given in Table 2 has the reducts *{$A_1$, $A_3$, $A_5$}*, *{$A_1$, $A_3$, $A_4$}* and *{$A_1$, $A_2$, $A_3$}* from which the one classifying the dataset with the highest accuracy is the reduct consisting of the attributes $A_1$, $A_3$ and $A_5$

## V. EXPERIMENTAL RESULTS

To estimate the performance of the proposed method, we compared the results generated by SR with the results generated by OAS for 9 datasets. In the experiments, we used a target machine with an Intel Core2Quad@2.83 GHz processor and 4 GB memory, running on Microsoft Windows 7 OS. The different datasets used in the experiments are from UCI repository with different characteristics such as: the number of attributes, the number of classes, the number of distinct values of the attributes and the number of examples. Firstly, with using decision tree algorithm the GDA of the datasets obtained (Chapter III). Secondly, the GDARs of the dataset are obtained by using discernibility function based feature selection program [39] (Chapter IV). Lastly, the classification tests are made by using Orange Data-Mining program [40]. We can see these datasets characteristics, GDAs and SRs in table 5 as following:

TABLE 5. THE CHARACTERISTICS OF THE SEVERAL DATASETS USED IN THE EXPERIMENTS

| Dataset | The number of | | GDA | SR | Number of attributes in the | | |
|---|---|---|---|---|---|---|---|
| | Examples | Classes | | | GDA | SR | OAS |
| Lymphography | 148 | 4 | a1a2a6a13a14 | a1a2a5a6a8a13a14a15 | 5 | **8** | 18 |
| Zoo | 101 | 7 | a9a12a13 | a4a6a9a12a13 | 3 | **5** | 16 |
| Breast Can. Wisconsin | 699 | 2 | a1a2a6 | a1a2a6a8 | 3 | **4** | 9 |
| Tictoctoe | 958 | 2 | a3a5a7 | a1a2a3a5a6a7a8a9 | 3 | **8** | 9 |
| Chess | 3196 | 2 | a1a6a10a15 a21a32a33a35 | a1a3a4a5a6a7a10a12a13a15a16a17a18a20a21 a22a23a24a25a26a27a28a30a31a32a33a34a35a36 | 8 | **29** | 36 |
| Voting | 435 | 2 | a3a4a11 | a1a2a3a4a6a9a11a13a14a16 | 3 | **10** | 16 |
| Monk | 556 | 2 | a1a2a5 | a1a2a5 | 3 | **3** | 6 |
| Solar Flare | 323 | 3 | a1a2a6 | a1a2a3a4a5a6a9a10a11a12 | 3 | **10** | 12 |
| Primary Tumor | 339 | 21 | a2a4a5a7a9a14 | a1a2a3a4a5a7a8a9a10a11a12a13a14a15a16a17 | 6 | **16** | 17 |

From Table 5, we can see that the average of the SR size is approximately 67% lesser than average of the OAS size. Thus, the classification algorithms can work faster and using less memory.

As classification algorithms, we used the algorithms *C4.5* [30], *K-NN* [41] with 7 neighbors and *Naïve Bayes* [42-46]. For estimating the classification accuracy of the algorithms, we used the most widely used cross-validation method [47]. More specifically, we used ten-fold cross-validation in which the dataset to be processed is permuted and partitioned equally into ten disjoint sets $D_1$, $D_2$, …, $D_{10}$. In each phase of a cross-validation, one of the yet unprocessed sets was tested, while the union of all remaining sets was used as training set for classification by the algorithms *C4.5*, *K-NN* and *Naive Bayes*. We obtained the classification accuracy of the algorithms for a certain dataset as average of the accuracies of the mentioned ten phases.

TABLE 6. THE CLASSIFICATION ACCURACY FOR THE DATASETS PROVIDED BY ORIGINAL ATTRIBUTE SETS AND BY SR

| Dataset | Classification Accuracy | | | | | |
|---|---|---|---|---|---|---|
| | K-NN | | Naive Bayes | | C4.5 | |
| | OAS | SR | OAS | SR | OAS | SR |
| Lymphography | 0.818 | **0.843** | 0.803 | **0.830** | 0.749 | **0.803** |
| Zoo | 0.962 | **0.980** | 0.912 | **0.990** | 0.971 | **0.980** |
| Breast Can. Wisconsin | 0.947 | **0.974** | 0.963 | **0.967** | 0.949 | **0.973** |
| Tictoctoe | 0.804 | **0.824** | 0.703 | **0.758** | **0.865** | 0.853 |
| Chess | 0.906 | **0.938** | 0.879 | **0.900** | **0.998** | 0.994 |
| Voting | 0.922 | **0.947** | 0.892 | **0.938** | 0.945 | **0.957** |
| Monk | 0.819 | **1.000** | 0.718 | **0.746** | 0.964 | **1.000** |
| Solar Flare | 0.666 | **0.681** | 0.672 | **0.688** | 0.712 | **0.785** |
| Primary Tumor | 0.372 | **0.405** | 0.493 | **0.536** | 0.404 | **0.469** |

From Table 6, we can find the classification performances of the classification algorithms (the higher accuracies of the classification algorithm are shown in bold). The SRs achieved better results in the 25 of the 27 experiments. The classification results of the *K-NN*, *Naïve Bayes* and *C4.5* algorithms are increased average 5.21%, 4,52% and 3,40%, respectively, for 9 datasets with using SR.

Consequently, in the proposed method SR is used instead of OAS. SR is the simplified form of the OAS that includes dominant attributes of the OAS. These SRs are preferred and performed at the classification stage because the following benefits: better classification accuracy, less working CPU time and less using memory for classification algorithms. The attributes in SRs are relevant attributes not redundant so the classification algorithms can create more effective rules and achieve better classification accuracies mostly.

## VI. CONCLUSION

In this paper a novel hybrid method for fast finding the RWBCA is proposed. Usually a dataset has a lot of reducts from which the only one can provide the best classification of this dataset. Unfortunately, finding and testing all of them is NP hard problem. Therefore, we propose a two-phase hybrid approach for obtaining the RWBCA. In the first phase, we obtain the group of the dominant attributes for the given dataset. Since this group is usually not sufficient for acceptable classification accuracy of the datasets, in the second phase we complete the obtained group of the attributes up to the RWBCA. In first and second phases we use the decision tree and discernibility function techniques, respectively. Such an approach allows us to reduce the reduct search space in a large scale. For instance, for some datasets with hundreds reducts we reduced the search space up to several reducts that obtained very quickly. We tested the performance of our approach with the popular algorithms such as K-NN, Naive Bayes and C4.5. We get better classification accuracies with the SRs in the 25 of 27 experiments.

## REFERENCES

[1] J.R. Quinlan, “Induction of decision trees,” Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.

[2] J.R. Quinlan, RM. Cameron-Jones “Induction of logic programs: Foil and related systems,” New Generation Computing, vol. 13 no. 3-4, pp. 287-312, 1995.

[3] Y. Kusunoki, M. Inuiguchi, J. Stefanowski, “Rule induction via clustering decision classes,” International Journal of Innovative Computing Information and Control, vol. 4, no. 10, pp.2663-2677, 2008.

[4] J.A. Jakubczyc, “The ant colony algorithms for rule induction,” Proceedings of AIML 05 Conference, CICC, Cairo, Egypt, pp. 112-117, 19-21 December 2005.

[5] J.W. Grzymala-Busse, Chien Pei B., "Classification methods in rule induction," Proceedings of the Fifth Intelligent Information Systems Workshop, Deblin, Poland, pp. 120–126, June 2–5 1996.
[6] R.S. Michalski, "On the Quasi-Minimal solution of the covering problem," Proceedings of the Fifth International Symposium on Information Processing, Bled, Yugoslavia, A3 (Switching Circuits), pp.125-128, 8-11 October 1969.
[7] R.S. Michalski, I. Mozetic, J. Hong, N. Lavrac, "The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains," Proc. AAAI, pp. 1041-1047, 1986.
[8] W. Cohen, "Fast effective rule induction," Proceedings of the 12th International Conference on Machine Learning, p.115-123, 1995.
[9] S.J. Hong, "R-MINI: An iterative approach for generating minimal rules from examples," IEEE Transactions on Knowledge and Data Engineering, vol. 9, no. 5, pp. 709-717, 1997.
[10] M. Muselli, D. Liberati, "Binary rule generation via hamming clustering," IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 6, pp. 1258-1268, 2002.
[11] N. Cercone, A. An, C. Chan, "Rule-induction and case-based reasoning: Hybrid architectures appear advantageous," IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 1 pp. 166-174, 1999.
[12] P. Smyth, R.M. Goodman, "An Information Theoretic Approach to Rule Induction from Database," IEEE Transactions on Knowledge and Data Engineering vol. 4 no. 4, pp. 301-316, 1992.
[13] D.S. Zhang, L. Zhou, "Discovering Golden Nuggets: Data Mining in Financial Application," IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews vol. 34, no.4, pp. 513-522, 2004.
[14] R. Kohavi, J.R. Quinlan, "Decision-tree discovery," Handbook of Data Mining and Knowledge Discovery, pp. 267-276, 2002.
[15] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, "Classification and Regression Trees," CA, Wadsworth, 1984.
[16] D.D. Patil, V.M. Wadhai, J.A. Gokhale, "Evaluation of Decision Tree Pruning Algorithms for Complexity and Classification Accuracy," International Journal of Computer Applications, vol. 11, no.2, pp. 23-30, 2010.
[17] J. Komorowski, L. Polkowski, A. Skowron, "Rough Set: A Tutorial," http:// folli.loria.fr/cds/1999/library/pdf/ skowron.pdf.
[18] A.E. Hassanien, J.M.H. Ali, "Rough set approach for generation of classification rules of breast cancer data," Informatica, vol. 15, no. 1, pp. 23–38, 2004.
[19] J.Y. Guo, V. Chankong, "Rough set-based approach to rule generation and rule induction," International Journal of General Systems, vol. 31, no. 6, pp. 601–617, 2002.
[20] J.F. Liu, Q.H. Hu, D.R. Yu, "A weighted rough set based method developed, for class imbalance learning," Information Sciences, vol. 178, no.4, pp. 1235–1256, 2008.
[21] E. Xu, S.C. Tong, L.S. Shao, Y. Li, D. Jiao, "Rough Set Research on Rule Extraction in Information Table," Proceedings of Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 208-212, 2007.
[22] M.A. Hall, G.Holmes, "Benchmarking Attribute selection techniques for discrete class data set mining," IEEE Transactions on Knowledge and data set Engineering vol. 15, pp. 6 pp. 1437-1447, 2003.
[23] H. Liu, L.Yu, "Toward integrating feature selection algorithms for classification and clustering," IEEE Transactions on Knowledge and data set Engineering, vol. 17, no. 4, pp. 491-502, 2005.
[24] A. Hassanien, "Fuzzy rough sets hybrid scheme for breast cancer detection," Image and Vision Computing, vol. 25, no. 2, pp. 172-183, 2007.
[25] J.Y. Wang, J. Zhou, "Research of reduct features in the variable precision rough set model," Neuro computing, vol. 72, no. 10-12 pp. 2643-2648, 2009.
[26] J. Wroblewski, "Ensembles of Classifiers Based on Approximate Reducts," Fundamenta Informaticae vol.47 no.3-4, pp. 351–360 2001.
[27] R. Jensen, Q. Shen, "Semantics-preserving dimensionality reduction: Rough and fuzzy-rough based approaches," IEEE Transactions on Knowledge and Data Engineering vol.16, no.12, pp.1457–1471, 2004.
[28] Z. Pawlak, "Rough Sets: Theoretical aspects of reasoning about data," Kluwer Academic Publishers, Boston, pp. 1-252, 1991.
[29] J.R. Quinlan, "Learning efficient classification procedures and their application to chess end games," Machine Learning: An Artificial Intelligence Approach, vol. 1, pp.463-482, 1983.
[30] J.R. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, 1993.
[31] P. Clark, T. Niblett, "The CN2 induction algorithm," Machine Learning, vol. 3, no.4, pp. 261-283, 1989.
[32] L.H. Wang, G.F. Wu, "Attribute Reduction and Information Granularity," 6th World Multi conference on Systemics, Cybernetics and Informatics, Vol I, Proceedings - Information Systems Development I, pp. 32-37, 2003.
[33] R. Jensen, Q. Shen, "Rough set based feature selection: A review," htp//cadair.aber.ac.uk/dspace/handle/2160/490, 52 pages, 2007.
[34] R.W. Swiniarski, A.Skowron, "Rough set methods in feature selection and recognition," Pattern Recognition Letters, vol. 24, no. 6, pp. 833–849, 2003.
[35] J. Wei, S. Wang, M. Wang, J. You, D. Liu, "Rough set based approach for inducing decision trees," Knowledge based systems, vol. 20, no. 8, 695-702, 2007.
[36] A. Skowron, C. Rauszer, "The discernibility matrices and functions in information systems," Fundamenta Informaticae, vol. 15, no. 2, pp. 331-362, 1992.
[37] S. Kahramanli, M. Hacibeyoglu, A. Arslan, "A Boolean Function Approach to Feature Selection in Consistent Decision Information Systems," Expert Systems with Application, vol. 38, no. 7, pp. 8229-8239, 2011.
[38] S. Kahramanli, M. Hacibeyoglu, A. Arslan, "Attribute Reduction by Partitioning the Minimized Discernibility Function," International Journal of Innovative Computing Information and Control, vol. 7, no. 5A, pp. 2167-2186, 2011.
[39] http://research.konya.edu.tr/dfbfs/
[40] http://orange.biolab.si/
[41] G. Shakhnarovish, T Darrell, P. Indyk, "Nearest-Neighbor Methods in Learning and Vision," MIT Press, 2005.
[42] T.Mitchel, "Machine Learning", McGraw-Hill, 1997.
[43] P. Minvielle, A. Doucet, A. Marrs, S. Maskell, "A Bayesian approach to joint tracking and identification of geometric shapes in video sequences", Image and Vision Computing, vol.28, no.1, pp.111-123, 2010.
[44] C. Pozna, R.-E. Precup, J. K. Tar, I. Skrjanc, S. Preitl, "New results in modelling derived from Bayesian filtering," Knowledge-Based Systems, vol. 23, no. 2, pp. 182-194, 2010.
[45] F. Zhang, W.-F. Xue, X. Liu, "Overview of nonlinear Bayesian filtering algorithm," Procedia Engineering, vol.15, pp. 489-495, 2011.
[46] G.E. D'Errico, "A la Kalman filtering for metrology tool with application to coordinate measuring machines," IEEE Transactions on Industrial Electronics, vol. 59 , no. 11, pp. 4377-4382, 2012.
[47] N. Mastrogiannis, B. Boutsinas, I. Giannikos, "A method for improving the accuracy of data mining classification algorithms," Computers & Operations Research, vol. 36, no. 10, pp. 2829-2839, 2009.