Engineering Applications of Artificial Intelligence 35 (2014) 71–83

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Improved decision tree construction based on attribute selection and data sampling for fault diagnosis in rotating machines

Nour El Islem Karabadji [a,*], Hassina Seridi [a], Ilyes Khelf [b], Nabiha Azizi [a], Ramzi Boulkroune [c]

[a] Electronic Document Management Laboratory (LabGED), Badji Mokhtar-Annaba University, P.O. Box 12, 23000 Annaba, Algeria
[b] Laboratoire de Mécanique Industrielle, Badji Mokhtar-Annaba University, P.O. Box 12, 23000 Annaba, Algeria
[c] URASM-CSC/ ANNABA Unité de recherche Appliquée en Sidérurgie Métallurgie Centre National de Recherche Scientifique et Technique en Soudage et Contrôle, Algeria

ARTICLE INFO

Article history:
Received 2 January 2014
Received in revised form
16 June 2014
Accepted 16 June 2014
Available online 8 July 2014

Keywords:
Decision tree construction
Pruning
Research graph
Attribute selection
Data sampling

ABSTRACT

This paper presents a new approach that avoids the over-fitting and complexity problems suffered in the construction of decision trees. Decision trees are an efficient means of building classification models, especially in industrial engineering. In their construction phase, the two main problems are choosing suitable attributes and database components. In the present work, a combination of attribute selection and data sampling is used to overcome these problems. To validate the proposed approach, several experiments are performed on 10 benchmark datasets, and the results are compared with those from classical approaches. Finally, we present an efficient application of the proposed approach in the construction of non-complex decision rules for fault diagnosis problems in rotating machines.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

In the industrial field, the risks of failure and disruption are increasing with the complexity of installed equipment. This phenomenon affects product quality, causes the immediate shutdown of a machine, and undermines the proper functioning of an entire production system. Rotating machines are a major class of mechanical equipment, and need the utmost care and continuous monitoring to ensure optimal operation. Traditionally, vibration analyses and many signal processing techniques have been used to extract useful information for monitoring the operating condition. Khelf et al. (2013) analysed the frequency domain to extract information and diagnose faults. Cepstral analysis has been used to construct a robust gear fault indicator (Badaoui et al., 2004), and a short-time Fourier transform representation was derived (Mosher et al., 2003). Other techniques have also been employed, such as the Wigner–Ville distribution (Baydar and Ball, 2001), continuous wavelet analysis (Kankar et al., 2011), and discrete wavelet analysis (Djebala et al., 2008).

Classification algorithms can be used in the construction of condition-monitoring diagnostic systems. For example, neural networks (Chen and Chen, 2011), support vector machines (Deng et al., 2011), and Bayesian classifiers (Yang et al., 2005) have all been applied. However, decision tree techniques are still preferred in engineering applications, because they allow users to easily understand the behaviour of the built models against the above-mentioned classifiers. Their use in such applications has been reported in numerous research papers, e.g. Sugumaran and Ramachandran (2007), Zhao and Zhang (2008), Sakthivel et al. (2010), and Sugumaran et al. (2007).

The construction of a decision tree (DT) includes growing and pruning stages. In the growing phase, the training data (samples) are repeatedly split into two or more descendant subsets, according to certain split rules, until all instances of each subset wrap the same class (pure) or some stopping criterion has been reached. Generally, this growing phase outputs a large DT that includes the learning examples and considers many uncertainties in the data (particularity, noise and residual variation). Pruning approaches based on heuristics prevent the over-fitting problem by removing all sections of the DT that may be based on noisy and/or erroneous data. This reduces the complexity and size of the DT. The pruning phase can under-prune or over-prune the grown DT. Moreover, many existing heuristics are very challenging (Breiman et al., 1984; Niblett and Bratko, 1987; Quinlan, 1987), but, unfortunately, no single method outperforms the others (Mingers, 1989; Esposito et al., 1997).

In terms of growing phase problems, there are two possible solutions: the first reduces DT complexity by reducing the number of learning data, simplifying the decision rules (Piramuthu, 2008).

[*] Corresponding author.
E-mail addresses: karabadji@labged.net (N.E.I. Karabadji), seridi@labged.net (H. Seridi), ilyeskhelf@gmail.com (I. Khelf), azizi@labged.net (N. Azizi), ramzi86@hotmail.com (R. Boulkroune).

http://dx.doi.org/10.1016/j.engappai.2014.06.010
0952-1976/© 2014 Elsevier Ltd. All rights reserved.

The second solution uses attribute selection to overcome over-fitting problems (Yildiz and Alpaydin, 2005; Kohavi and John, 1997). To overcome both the DT size and over-fitting risks, we propose to combine attribute selection and data reduction to construct an Improved Unpruned Decision Tree $\mathcal{IUDT}$. The optimal DT construction (DTC) problem will thus be converted into an exploration of the combinatorial graph research space problem. The key feature of this proposition is to encode each subset of attributes A_i and a samples subset X_j into a couple (A_i, X_j). All possible (A_i, X_j) couples form the research space graph. The results show that the proposed schematic largely improves the tree performance compared to standard pruned DTs, as well as those based solely on attribute selection or data reduction.

The rest of the paper is organized as follows: In Section 2, some previous studies on DTC are briefly discussed. Section 3 introduces the main notions used in this work. In Section 4, we describe our approach based on attribute selection and database sampling to outperform conventional DTC. Section 5 reports the experimental results using 10 benchmark datasets. In Section 6, $\mathcal{IUDT}$ is applied to the problem of fault diagnosis in rotating machines. Finally, Section 7 concludes the study.

2. Related work

This section describes post-pruning approaches that have been proposed to improve DTC. Their common aim was to decrease (1) the tree complexity and (2) the error rate of an independent test dataset. Pruning methods have various differences that can be summarized as follows:

1. the necessity of the test dataset;
2. the generation of a series of pruned sub-trees or the processing of a single tree;
3. the pruning determination criteria.

Breiman et al. (1984) developed error-complexity pruning, which uses the cost-complexity risk. The pruning measure uses an error rate penalty based on the sub-tree size. The errors and the size of the tree's leaves (complexity) are both considered in this pruning method. The cost-complexity risk measurement of all possible sub-trees in an initial DT T_0 is calculated as the training error $R(t)$ added to the product of a factor α and the number of leaves $|\overline{t}|$ in the sub-tree t, i.e. $RC_\alpha(t) = R(t) + \alpha(|\overline{t}|)$. A series of sub-decision trees with the smallest value of α are selected to be pruned. Finally, the correctly pruned sub-tree t is selected from the α sequence of sub-trees using an independent test dataset. The final selection is based on the error rate or standard error (assuming a binomial distribution).

Reduced-error pruning, proposed by Quinlan (1987), produces a series of pruned DTs using the test dataset. A complete DT T_0 is first grown using the training dataset. A test dataset is then used, and for each node in T_0, the number of classification errors made on the pruning set when the sub-tree t is kept is compared with the number of classification errors made when t is turned into a leaf. Next, the positive difference between the two errors is assigned to the sub-tree root node. The node with the largest difference is then pruned. This process is repeated until the pruning increases the misclassification rate. Finally, the smallest version of the most accurate tree with respect to the test dataset is generated.

In contrast to reduced-error pruning, the necessity of separate test datasets can be avoided using pessimistic error pruning (PEP, Quinlan, 1987). This uses the binomial continuity correction rate to obtain a more realistic estimate of the misclassification rate. The misclassification correction depends on the number of leaves and misclassifications.

Error-based pruning (EBP, Quinlan, 1993) is an improved version of PEP that traverses the tree according to a bottom-up post-order strategy. No pruning dataset is required, and the binomial continuity correction rate of PEP is used. Therefore, the difference is that, in each iteration, EBP considers the possibility of grafting a branch t_y in place of the parent of y itself. The estimation errors t_x, t_y are calculated to determine whether it is convenient to prune node x (the tree rooted by x replaced by a leaf), replace it with t_y (the largest sub-tree), or keep the original t_x.

Recently, Luo et al. (2013) developed a new pruning method based on the structural risk of the leaf nodes. This method was developed under the hypothesis that leaves with high accuracies mean that the tree can classify the training data very well, and a large volume of such leaves implies generally good performance. Using this hypothesis, the structural risk measures the product of the accuracy and the volume of leaf nodes. As in common pruning methods, a series of sub-trees are generated. The process visits each node x on DT T_0 (t_x is a sub-tree whose root is x). For each sub-tree t_x, feasible pruning nodes are found (their two children are leaves), and the structural risks are measured. Finally, the sub-tree that maximizes the structural risk is selected for pruning.

Additional post-pruning methods have been proposed, such as critical value pruning (Mingers, 1987), minimum error pruning (Niblett and Bratko, 1987), and DI pruning (which balances both the Depth and the Impurity of nodes) (Fournier and Crémilleux, 2002). The choice of DT has also been validated (Karabadji et al., 2012), and genetic algorithms used to pull out the best tree over a set of different models (e.g. BFTree, J48, LMT, Hall et al., 2009). To select the most robust DT, all models were generated and their performances measured on distinct training and validation sets. In this work, the main objective is to construct DTs without under-pruning or over-fitting the training dataset, and without choosing between different pruning methods. Two prior works have shown that unpruned DTs give similar results to pruned trees when a Laplace correction is used to calculate the class probabilities (Bradford et al., 1998; Provost and Domingos, 2003).

The identification of smaller sets of highly predictive attributes has been considered by many learning schemes. Attribute selection shares the same objective as pruning methods, namely the elimination of irrelevant, redundant, and noisy attributes in the building phase to produce good DT performance. Many studies have investigated and improved classification models (Bermejo et al., 2012; Macaš et al., 2012). In these works, wrapper techniques have been applied to attribute selection. A target learning algorithm is used to estimate the value of attribute subsets. The process is driven by the binary relation "$\subseteq$" between attribute subsets. The search process can be conducted on a depth-first or breadth-first basis, or a combination of both (e.g. "A star" (A*) algorithm). Wrappers are generally better than filters, but the improved performance comes at a computational cost—in the worst case, 2^m subsets of attributes must be tested (m is the number of attributes) (Kohavi and John, 1997).

Similar to attribute selection, DTs can be improved by reducing the data complexity, as well as reducing the effects of unwanted data characteristics. Data reduction essentially involves dimensionality reduction and/or example reduction (Piramuthu, 2008). Generally, reduction methods use sampling (e.g. random, stratified) to select examples for consideration in the learning phase (Ishibuchi et al., 2001; Liu, 2010).

In conclusion, different pruning techniques have been studied, but none is adequate for all varieties of problem. There has been a recent focus on attribute selection and sampling data to improve DTC. To realize a better DT for a specific application, we propose the $\mathcal{IUDT}$ algorithm, which combines a novel scheme of random database sampling with an attribute selection wrapper process. The main objective of our study is to reduce the effective number of examples and training data attributes, and thus minimize the size of the DT.

3. Preliminaries

Before describing our approach, we give some basic results on the ordering of sets, classifier over-fitting problems, attribute relevance and redundancy problems, and finally DTC. The definitions presented in this section use the same semantics as in Davey (2002).

3.1. Partial order

An ordered set is a collection of elements with an order relation.

Definition 1. A binary relation R on set E is a partial order if it is reflexive, transitive, and anti-symmetric.

- $\forall x \in E$, xRx (reflexivity),
- $\forall (x,y) \in E \times E$, $(x\ R\ y\ et\ y\ R\ x) \Longrightarrow (x=y)$ (anti-symmetry),
- $\forall (x,\ y,\ z) \in E \times E \times E, (x\ R\ y\ et\ y\ R\ z) \Longrightarrow (x\ R\ z)$ (transitivity).

Example 1. Let $X=\{a,b,c,d,e\}$ and $P=(X,\preccurlyeq)$ be an ordered set, where $\preccurlyeq$ defines the following order on X: $\preccurlyeq = \{(a,b), (a,e), (c,b), (c,d), (c,e), (d,e), (a,a), (b,b), (c,c), (d,d), (e,e)\}$.

We can represent an ordered set P as an oriented graph whose nodes correspond to the X elements and whose edges denote the relation $\preccurlyeq$, with loops representing the couples (x,x). Fig. 1 illustrates Example 1.

Definition 2 (*Graph*). A graph $G=(V,E)$ consists of a set of vertices V and an edges set $E \subseteq V \times V$. Each edge $e \in E$ is associated with an unordered pair of vertices.

In the case of an oriented graph, each edge $e \in E$ is associated with an ordered pair of vertices. In the rest of this section, we denote a research graph L and an element in L (i.e. a vertex $v \in G$) as a pattern.

3.1.1. Specialization and generalization patterns

Specialization (generalization) is a binary relation that defines a partial order $\preccurlyeq$ on the patterns in L. A pattern φ is more general than another pattern θ if $\varphi \preccurlyeq \theta$. Similarly, θ is more specific than φ, i.e. for the relation $a \preccurlyeq e$ in Example 1, a is more general than e and e is more specific than a.

3.1.2. Specialization and generalization operators

Let L be a partially ordered set of patterns. The specialization operator P_s associates each pattern $\varphi \in L$ with a pattern set that is more specific: $P_g(\varphi)=\{\theta \in L | \varphi \prec \theta\}$. Similarly, we can define a generalization operator P_g such that $P_g(\varphi)=\{\theta \in L | \theta \prec \varphi\}$. An operator P_s is said to be direct (immediate) if it only associates φ with the most general patterns of the set of patterns that are more specific $P_s(\varphi)$, $P_s(\varphi)=\min(P_s(\varphi))$. Then, an operator P_g is said to be direct (immediate) if it only associates φ with the most specific patterns of the set of patterns that are more general $P_g(\varphi)$, $P_g(\varphi)=\max(P_g(\varphi))$.

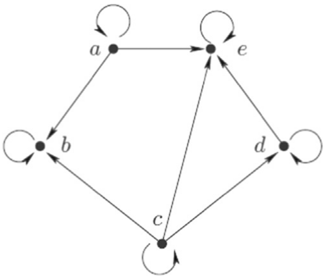

Fig. 1. Partial order.

3.1.3. Minimum and maximum patterns

Let be a specialization relation in L, and $\Phi \subseteq L$ be a patterns set. $\min(\Phi)$ is the most general patterns set of Φ, and $\max(\Phi)$ is the most specific patterns set of Φ.

$$\min(\Phi)=\{\varphi \in \Phi | \nexists \theta \in \Phi \text{ s.t. } \theta \prec \varphi\}$$

$$\max(\Phi)=\{\varphi \in \Phi | \nexists \theta \in \Phi \text{ s.t. } \varphi \prec \theta\}$$

Using a direct order relation, we can represent a partially ordered set by a directed graph and acyclic Hasse diagram. Fig. 2 illustrates Example 1 as a Hasse diagram.

3.1.4. Research graph traversing strategy

Graph specialization is generated by the specialization order relation defined on L, which can be traversed in several modes:

- Breadth-first search: traverses the research space (specialization graph) elements in a bottom-up manner and generates patterns level-by-level. Known to be an apriori-like where all motifs with the same level are explored before the more specific ones.
- Depth-first search: achieves the quickest possible solution by exploring the immediate successor of any generated pattern, and specializes as much as possible the pattern level before exploring patterns of the same level.

3.2. Over-fitting

Consider the sample data illustrated in Fig. 3. Let us assume that “circles” and “plus signs” in this figure correspond to sampled

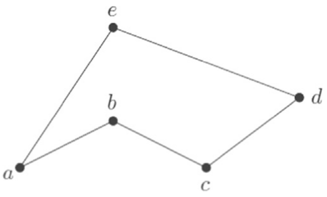

Fig. 2. Hasse diagram.

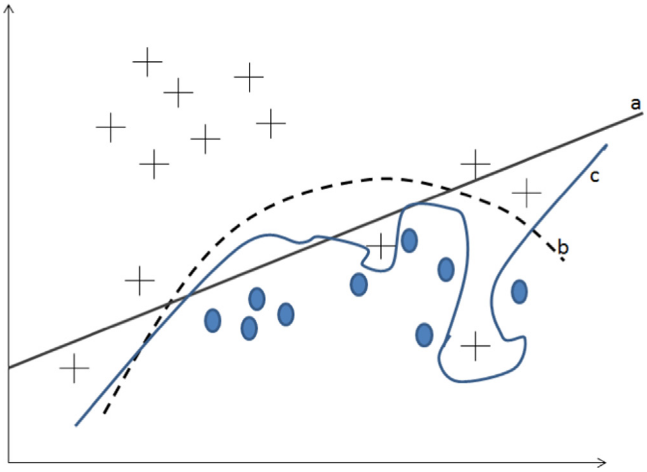

Fig. 3. Over-fitting in decision tree learning. (a) An over-generalization. (b) An ideal classification. (c) An over-fitting.

observations from two classes and from (a, b, c) different classifier models.

Classifier (a) simply classifies the data according to a straight line. This gives a very poor classification, considered as a hazard classification. In contrast, classifier (c) over-fits the training data, becoming increasingly dependent on the samples. Thus, its capacity to correctly predict other data classes decreases. Finally, classifier (b) gives the best generalization of the learning process, and has the smallest probability of misclassifying new data.

Definition 3 (*Over-fitting*). $h \in H$ over-fits training set S if there exists $h' \in H$ that has a higher training set error but lower test error on the test data. (More specifically, if learning algorithm A explicitly considers and rejects h' in favour of h, we say that A has over-fitted the data.)

3.3. Attribute particularity

The identification (elimination) of relevant (redundant) attributes is the main purpose of an attribute selection algorithm.

3.3.1. Relevance

In machine learning, relevance includes three disjoint categories: strong relevance, weak relevance, and irrelevance (Kohavi and John, 1997), in order of importance. Strongly relevant attributes should be conserved by any attribute selection algorithm; however, Ruiz et al. (2006) state that there is no guarantee that a feature will necessarily be useful to an algorithm just because of its relevance (or vice versa). Weakly relevant attributes could be conserved or not, depending on the evaluation measure (e.g. accuracy, simplicity) and other selected attributes. Irrelevant attributes should be eliminated.

3.3.2. Incremental relevance

Caruana and Freitag (1994) define incremental relevance by considering the monotonicity of the accuracy and order of the set of subsets $P(R, \subseteq)$.

Definition 4 (*Incremental usefulness*). “Given data D, a learning algorithm T, and a subset of attributes X, the attribute e is incrementally useful to T with respect to X if the accuracy of the hypothesis that T produces using the group of attributes $\{e\} \cup X$ is better than the accuracy achieved using just the subset of attributes X” (Caruana and Freitag, 1994).

To obtain a predictive feature subset of attributes, the above definition is useful in the present work.

3.3.3. Redundancy

Attribute redundancy in machine learning is widely accepted as the correlation between attributes. Two attributes are redundant to each other if their values are completely correlated. Correlation between two variables can be checked based on the entropy, or the random variable uncertainty (Xing et al., 2001; Liu and Yu, 2005).

3.4. Decision trees

DTs are built recursively, as illustrated in Fig. 4, following a top-down approach. They are composed of a root, several nodes, branches, and leaves. DTs grow according to the use of an attribute sequence to divide training examples into n classes. Tree building can be described as follows. First, the indicator that ensures the best split of the training examples is chosen, and population subsets are distributed to new nodes. The same operation is repeated for each node (subset population) until no further split operations are allowed. Terminal nodes are composed of populations in the same class (increased proportion in some types of DT). The classification operation assigns an individual to a terminal node (leaf) by satisfying the set of rules oriented to this leaf. The set of rules forms the DT.

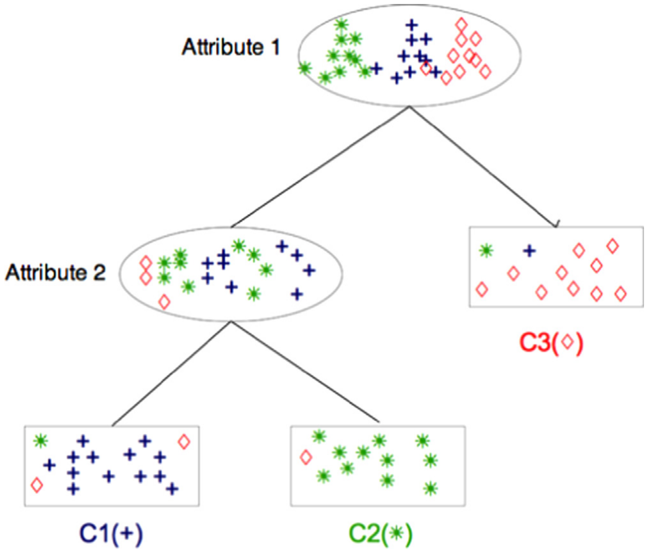

Fig. 4. Decision tree example.

It is clear that the size and uncertainty of the training examples are the central issue in DTC, and so we aim to make it less complex while retaining high performance. DT performance is mainly based on determining its size (Breiman et al., 1984). It has been proved that the tree size grows with the number of training data observations (Oates, 1997).

4. The $\mathcal{IUDT}$ approach

This section describes the idea of replacing the post-pruning phase by attribute selection and dataset reduction. The main objective is to illustrate that the performance of an unpruned DT can be improved using these two steps. Wrapper attribute selection can eliminate irrelevant and redundant attributes, and data reduction reduces the size complexity problem. We must therefore determine both the best attribute subset and the subset of training examples used to build the best unpruned decision tree t^*. The key features of the proposed method are as follows: (i) data preprocessing, (ii) definition of attribute combinations as a level-wise research space (specialization graph), and (iii) application of an oriented breadth-first exploration in the research space to find the best unpruned decision tree t^*. Fig. 5 presents a schematic of the proposed method.

4.1. Data preprocessing

Experimental analyses on DT pruning methods give the following split of training and test data: 25 instances of 70% training and 30% test data (Esposito et al., 1997), 9 instances of 60% training and 40% test data (Mingers, 1989), and 10 instances of 25% training and 75% test data (Bradford et al., 1998). Multiple instances were used to avoid the assumption that “a single random split of data may give unrepresentative results”. To obtain representative results in the experimentation stage, and to reduce the size of the target robust unpruned DT, the dataset was randomly split into a 50% training set and a 50% test set five times. Each training set was further split at random into three sub-training sets containing 50%

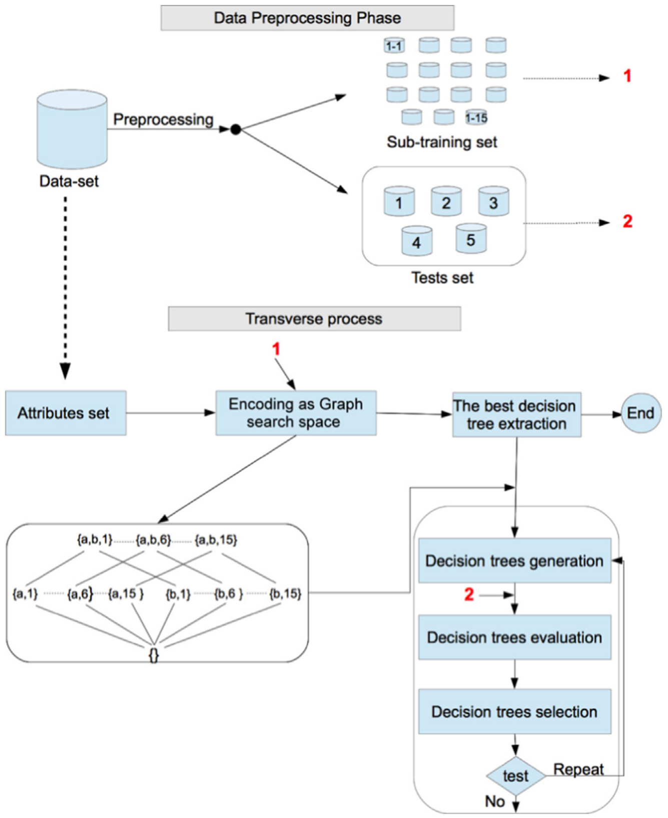

Fig. 5. Description of the $\mathcal{IUDT}$ approach.

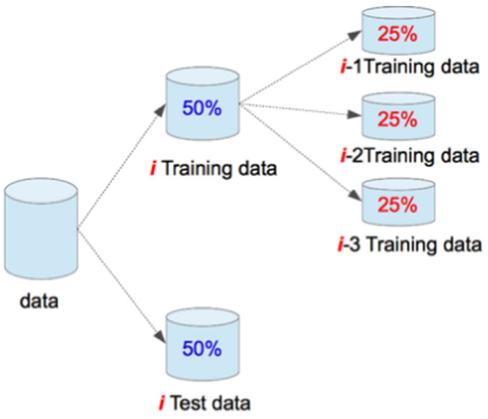

Fig. 6. Data preprocessing.

of the whole training set. This results in 15 training sets of 25% of the dataset, as in Bradford et al. (1998), but the test set remains at 50%. Note that for each of the five test sets, there are three sub-training sets. Fig. 6 illustrates the splitting process, where i varies from 1 to 5.

The chosen splitting process has been proposed to give a biased estimation prediction error. The principal justification is that the chosen process overlaps the test set because of the random sampling.

4.2. Encoding

We use a wrapper algorithm to select the best couple (X,i), where X is the attribute subset of the attribute set E and i is a sub-training set of the training set I. (X,i) constructs the best DT t^*, and the problem must be represented as a research graph. The research graph is an ordered set P of the couples set R according to the binary relation $\subseteq$. The use of the proposed data preprocessing step will increase the number of subsets traversed by the wrapper algorithm, i.e. 15×2^m. The set of couples R is defined as

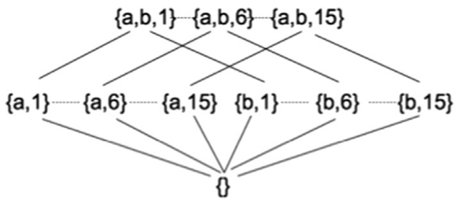

Fig. 7. Graph research space.

$$R = \{(X, i) | X \subseteq E, i \in I\} \quad (1)$$

Example 2. For a dataset that contains two attributes $\{a, b\}$, the couples set is represented as follows:

$$R = \{(a, 1), \ldots, (a, 15), (b, 1), \ldots, (b, 15), (a, b, 1), \ldots, (a, b, 15)\}$$

Fig. 7 illustrates the research graph $P\langle R, \subseteq \rangle$.

Formally, the task of finding t^* (the best unpruned DT) can be described as follows. Consider a database D, and a language LT for expressing R elements (couples (X,i)) which are used to build DTs. Let a set I of 15 sub-training sets, and a set S of five testing sets v, respectively, and, as well as an objective function F. The main problem is to extract t^* such that $t^* = \{t(X, i) \in LT | \max F(t)\}$. The objective function evaluates each constructed tree with i using the other learning sub-sets $\forall a \in S \backslash i$ and the b-test set. This is calculated by the function $w : I \rightarrow [1..5]$, where $b = w(i)$ with respect to the size of X. The function $w : I \rightarrow [1..5]$ is given by

$$w(c) = \begin{cases} 1 & \text{if } 1 \leq c \leq 3 \\ 2 & \text{if } 4 \leq c \leq 6 \\ 3 & \text{if } 7 \leq c \leq 9 \\ 4 & \text{if } 10 \leq c \leq 12 \\ 5 & \text{if } 13 \leq c \leq 15 \end{cases} \quad (2)$$

4.3. Research space exploration

A breadth-first search is adopted to traverse the graph. The proposed search method has similar characteristics to apriori-based frequent itemsets mining algorithms (Agrawal et al., 1994). The search for the best unpruned tree t^* starts with the empty subset $\emptyset$. Exceptionally, the size increases by two as we go from $\emptyset$ to couples containing one attribute and one sub-training index i. The search process then proceeds in a bottom-up manner. At each iteration, the size of the newly discovered subsets $(X, i) \in R$ increases by one with respect to the incremental relevance property (IRP), which is a predicate $IRP : L_{k+1} \rightarrow \{0, 1\}$. The new intermediate candidates L_{k+1} are generated by joining two similar but slightly different (by one attribute) subsets that have already been discovered, C_k. The new candidates C_{k+1} are the L_{k+1} that satisfy the proportional relevance property (PRP), which is a predicate $PRP : L_{k+1} \rightarrow \{0, 1\}$. The process is repeated iteratively, alternating between candidate generation and evaluation phases, until there are no new candidates in C_{k+1} ($C_{k+1} = \emptyset$). For each explored couple $(X, i) \in R$, a DT is built using only the subset of examples i that is divided over the attributes subset X, i.e. a particular permutation of the attributes in X. As illustrated earlier, the attributes sequence is ordered based on the split criteria (e.g. Gini index, impurity-based criteria, twoing criterion). At this stage, the proposed approach features two opportunities: a personalized DT model definition and a predefined model (e.g. BFTree, J48, REPTree, SimpleCart), which is subject to the constraint of avoiding the pruning phase. Every constructed DT is evaluated according to formula (3). Algorithm 1 gives the pseudo-code of the proposed traversal method.

Algorithm 1. | *The best unpruned DT search function.*

```
Input: A dataset $D$, a language $LT$, predicates $IRP$ and $PRP$.
Output: The optimal DT $t^*(X,i)$.
$L_1 = \{(e,i) | e \in E, i \in I\}$
$C_1 = \{(e,i) | (e,i) \in L_1, IRP(t(e,i))$ and $PRP(t(e,i))\}$
$t^* = Best-of(C_1)$
$i=1$
while $C_i$ is not empty do
    $L_{i+1} = \{(Y,i) | \forall (X,i) \in C_i, \forall e \in E, Y = X \cup e\}$
    $C_{i+1} = \{(X,i) | (X,i) \in L_{i+1}, IRP(t(X,i))$ and $PRP(t(X,i))\}$
    $t' = Best-of(C_{i+1})$
if $Accuracy(t^*) < Accuracy(t')$ then
    $t^* = t'$
end if
    $i=i+1$
end while
return $t^*$
```

Until recently, every wrapper algorithm that considered more than 30 attributes was computationally expensive. There are many methods of speeding up the traversal process. The principal concepts are based on minimizing the evaluation subsets (Hall and Holmes, 2003; Bermejo et al.; Ruiz et al., 2006), or using a randomized search (Stracuzzi and Utgoff, 2004; Huerta et al., 2006). To tackle this problem, the incremental relevance and a proportional of 5% of the best subset's accuracy properties are adopted. The accuracy of the DT constructed using couple (X,i) is calculated as follows:

$$Accuracy(t(X,i)) = average\left(\sum_{a \neq i}^{I} Accuracy(t(X,i),a) + Accuracy(t(X,i),w(i))\right). \tag{3}$$

Fig. 8 shows an instance of IRP and PRP. IRP aims to eliminate the generated candidates L_{k+1} by the addition of the gray and green attributes (i.e. IRP is not satisfied), and PRP aims to keep the attribute subsets that have an accuracy that is greater than that of the best of the intermediate candidates (L_{k+1}) minus 5% (e.g. on the right of Fig. 8, the black candidate is the best, white candidates are eliminated, and only the gray ones are kept).

5. Validation of $\mathcal{IUDT}$

The proposed method is implemented and tested on 10 standard machine learning datasets, which were extracted from the UCI collection (Blake and Merz, 1998). The code is implemented in Java using the WEKA framework (Hall et al., 2009) and GUAVA Google library (Bourrillion et al., 2010). Experiments were conducted and compared with the results from the original WEKA pruned DTs, i.e. **DTP** (*D*ecision *T*ree construction technique with use of *P*runing phase), an "attribute selection" algorithm implementation that behaves like a wrapper algorithm, i.e. **IUDTAS** (*I*mproved *U*npruned *D*ecision *T*ree construction using only *A*ttribute *S*election), and a second algorithm that considers only the sampling step, i.e. **IUDTSD** (*I*mproved *U*npruned *D*ecision *T*ree construction using only *S*ampling *D*ata).

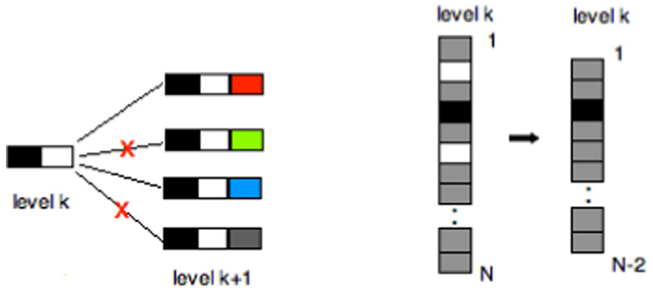

Fig. 8. *IRP*, *PRP* examples. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

The datasets are described in Table 1. The "% Base error" column refers to the percentage error obtained if the most frequent class is always predicted.

In addition, the selected datasets represent various applications. Three DTs, namely J48, SimpleCart, and REPTree, were chosen to apply different pruning methods within their standard WEKA implementation. The main DT characteristics considered in the experiments are reported in Table 2.

The experiments are based on the DTs reported in Table 2, implemented in their standard settings but without the pruning phase (unpruned DTs).

Tables 3, 4 and 5 list the classification accuracy for J48, REPTree, and SimpleCart, respectively, when the $\mathcal{IUDT}$ algorithm is applied to the 10 experimental datasets. The tables show that the number of attributes used to construct the DT is less than the number of data attributes, and only about one-sixth are used in the case of large attribute sets. The accuracy results show that the DTs outperform the *%-based accuracy* when only the most frequent class is continually predicted.

Table 6 shows WEKA's DTs built using the pruning phase (**DTP**). The design of these trees is mostly based on the construction (growing and pruning) phase, where 50% of the examples (instances) are used as the training set.

As in the proposed approach, each dataset was randomly split into training and test sets five times (50% split). The results reported in Table 6 represent the mean prediction accuracy over the five *i-test* datasets, which gives a comparison of accuracy and size between the proposed approach and the standard implementation **DTP**, i.e. using the pruning phase.

Table 6 shows that results differ from one model to another. Generally, the J48 trees are larger than the REPTree and SimpleCart DTs. This phenomenon is due to the pruning technique applied. In contrast to the *Reduced-error pruning* (REPTree) and *Error Complexity Pruning* (SimpleCart) techniques, the EBP technique employed by J48 (and discussed in Section 2) grafts one of the sub-trees of a sub-tree X that was selected to be pruned. Furthermore, the accuracy of the J48 DTs is the best for six datasets. Clearly, these results provide a picture of the over-pruning and under-pruning encountered in some databases. Over-pruning can be observed in the case of the Zoo and Breast-cancer datasets using REPTree and

Table 1
Main characteristics of the databases used for the experiments.

Dataset	*No. classes*	*No. attributes*	*No. instances*	*%Base error*
Zoo	7	17	101	59.40
Glass	7	9	214	35.52
Sonar	2	60	208	46.64
Ecoli	8	7	336	57.44
Diabetes	2	8	768	34.90
Hepatitis	2	19	155	20.65
Tic-tac-toe	2	9	958	34.65
Breast-cancer	2	9	286	29.72
Primary-tumor	21	17	339	75.22
Waveform-5000	3	40	5000	66.16

Table 2
Main characteristics of the DTs used for the experiments.

DTs	Split criteria	Pruning method	Principal standard options
J48 (C 4.5)	Gain ratio	Error-based	The confidence pruning is 0.25 The minimum number of instances at leaves is 2 One fold is used for pruning Consider the sub-tree raising operation when pruning
REPTree	Gain ratio	Reduced-error	The minimum number of instances at leaves is 2 One fold is used for pruning
SimpleCart	Gini index	Error-complexity	Binary split for nominal attributes The minimum number of instances at leaves is 2 One fold is used for pruning Five fold internal cross-validation

Table 3
$\mathcal{IUDT}$ approach applied to J48 results.

Data	# Attributes	Size	Accuracy
Zoo	3	11	90.50
Glass	5	29	70.56
Sonar	7	17	91.10
Ecoli	5	15	85.26
Diabetes	5	25	80.92
Hepatitis	6	15	84.09
Tic-tac-toe	8	73	84.70
Breast-cancer	5	51	79.02
Primary-tumor	10	42	53.40
Waveform-5000	7	179	83.80

Table 4
$\mathcal{IUDT}$ approach applied to REPTree results.

Data	# Attributes	Size	Accuracy
Zoo	5	13	95.00
Glass	5	27	76.40
Sonar	6	17	91.58
Ecoli	5	15	83.33
Diabetes	6	57	77.60
Hepatitis	3	9	79.22
Tic-tac-toe	9	76	81.83
Breast-cancer	5	68	78.49
Primary-tumor	10	40	56.30
Waveform-5000	6	295	83.08

Table 5
$\mathcal{IUDT}$ approach applied to SimpleCart results.

Dataset	# Attributes	Size	Accuracy
Zoo	5	13	95.00
Glass	4	19	72.66
Sonar	6	17	94.95
Ecoli	3	19	85.56
Diabetes	5	61	78.71
Hepatitis	6	17	89.93
Tic-tac-toe	9	49	93.05
Breast-cancer	6	31	77.92
Primary-tumor	10	43	52.51
Waveform-5000	6	231	82.19

Table 6
DTP results.

Dataset	J48		REPT		SCart	
	Size	Accuracy	Size	Accuracy	Size	Accuracy
Zoo	13	95.20	1	43.60	1	43.60
Glass	35	80.56	9	72.33	9	69.90
Sonar	19	89.42	3	77.88	9	80.76
Ecoli	25	82.26	13	82.26	15	82.02
Diabetes	31	81.40	39	80.57	5	77.55
Hepatitis	9	87.01	7	85.71	17	90.12
Tic-tac-toe	97	84.88	64	79.16	45	92.94
Breast-cancer	20	71.04	1	72.16	1	72.16
Primary-tumor	46	52.30	20	44.85	21	48.04
Waveform-5000	341	85.36	87	80.28	49	79.36

Table 7
IUDTSD results.

Dataset	J48		REPT		SCart	
	Size	Accuracy	Size	Accuracy	Size	Accuracy
Zoo	11	94.00	1	45.00	7	55.50
Glass	17	74.29	25	74.06	17	71.72
Sonar	11	85.09	9	84.61	9	83.89
Ecoli	17	84.11	9	84.22	29	84.52
Diabetes	27	81.70	45	81.90	47	82.29
Hepatitis	11	89.61	9	87.66	13	88.61
Tic-tac-toe	73	84.70	91	83.97	49	92.95
Breast-cancer	27	77.62	61	73.07	7	77.27
Primary-tumor	49	50.88	50	50.14	45	54.28
Waveform-5000	197	83.98	191	84.47	155	82.48

SimpleCart. An under-pruning case is illustrated in the case of the Ecoli dataset, where the accuracy of the J48 and REPTree models is the same when the size of J48 is much larger. Consequently, we can conclude that J48 is an under-pruned DT.

To demonstrate the effectiveness of the proposed method against an algorithm that only applies a sampling process without attribute selection, we generated DTs using the 15 sub-learning sets of examples *i*–*j train* ($i \in [1\ldots5]$ and $j \in [1\ldots3]$). The results reported in Table 7 give the mean prediction accuracy over the five *i*-*test* datasets. We can observe that, although the sizes of the trees are greater than the pruned DTs, their accuracy is much better. The accuracy of algorithms that only apply attribute selection without data sampling is reported in Tables 8–10. These results were obtained by implementing a wrapper algorithm that traverses the research graph space, as in the proposed method, using a random 50% subset (instances) as learning data. The best DTs validated by the same preprocessing output used in the proposed approach are selected (the five *i*-*test* datasets).

Tables 11–13 compare the results from $\mathcal{IUDT}$ based on a combination of sampling and attribute selection with the original WEKA pruned DT construction (Table 11), an approach that only applies data sampling, i.e. **IUDTSD** (Table 12), and an approach that only applies attribute selection i.e. **IUDTAS** (Table 13). The "+" symbol signifies that the proposed approach produces better results than the comparative method. Otherwise, we use the

Table 8
Application of **IUDTAS** to J48 results.

Dataset	# Attributes	Size	Accuracy
Zoo	5	11	93.60
Glass	2	27	79.81
Sonar	3	13	87.30
Ecoli	3	17	80.71
Diabetes	3	11	78.95
Hepatitis	3	9	85.45
Tic-tac-toe	5	124	84.63
Breast-cancer	2	12	75.94
Primary-tumor	8	38	53.72
Waveform-5000	6	189	81.21

Table 9
Application of **IUDTAS** to REPTree results.

Dataset	# Attributes	Size	Accuracy
Zoo	4	11	91.20
Glass	2	45	80.93
Sonar	3	27	87.11
Ecoli	3	53	84.52
Diabetes	3	89	84.79
Hepatitis	3	27	89.09
Tic-tac-toe	4	94	78.91
Breast-cancer	2	21	75.94
Primary-tumor	8	54	53.60
Waveform-5000	8	505	86.01

Table 10
Application of **IUDTAS** to SimpleCart results.

Dataset	# Attributes	Size	Accuracy
Zoo	5	11	94.40
Glass	2	47	82.80
Sonar	4	27	91.73
Ecoli	3	45	81.66
Diabetes	2	145	82.60
Hepatitis	3	27	89.09
Tic-tac-toe	6	105	90.43
Breast-cancer	2	17	75.94
Primary-tumor	7	61	55.14
Waveform-5000	8	233	86.41

Table 11
Comparison of DTP and $\mathcal{IUDT}$ results.

Dataset	J48		REPT		SCart	
	Size	Accuracy	Size	Accuracy	Size	Accuracy
Zoo	+	–	+	+	+	+
Glass	+	–	–	+	–	+
Sonar	+	+	–	+	–	+
Ecoli	+	+	–	+	–	+
Diabetes	+	=	–	–	–	+
Hepatitis	–	–	–	–	=	=
Tic-tac-toe	+	=	–	+	–	=
Breast-cancer	–	+	+	+	–	+
Primary-tumor	+	+	–	+	–	+
Waveform-5000	+	–	–	+	–	+

symbol "–". It is important to note that the over-pruned DTs of size one (i.e. Zoo and Breast-cancer) are always considered to be worse, and the accuracy is considered to be equal if the difference between the $\mathcal{IUDT}$ and the comparative approach (i.e. **DTP**, **IUDTSD**, or **IUDTAS**) is in the interval $[-1,+1]\%$. Otherwise, the tree with the greatest accuracy is considered to be better.

Table 12
Comparison of **IUDTSD** and $\mathcal{IUDT}$ results.

Dataset	J48		REPT		SCart	
	Size	Accuracy	Size	Accuracy	Size	Accuracy
Zoo	=	–	+	+	–	+
Glass	–	–	–	+	–	=
Sonar	–	+	–	+	–	+
Ecoli	+	+	–	=	+	+
Diabetes	+	=	–	–	–	–
Hepatitis	–	–	=	–	–	+
Tic-tac-toe	=	=	+	–	=	=
Breast-cancer	–	+	–	+	–	+
Primary-tumor	+	+	+	+	+	–
Waveform-5000	+	=	–	+	–	=

Table 13
Comparison of **IUDTAS** and $\mathcal{IUDT}$ results.

Dataset	J48		REPT		SCart	
	Size	Accuracy	Size	Accuracy	Size	Accuracy
Zoo	–	=	–	+	–	+
Glass	–	–	+	–	+	–
Sonar	–	+	+	+	+	+
Ecoli	+	+	+	–	+	+
Diabetes	–	+	+	–	+	–
Hepatitis	–	–	+	–	+	=
Tic-tac-toe	+	=	+	+	+	+
Breast-cancer	–	+	–	+	–	+
Primary-tumor	–	–	+	+	+	–
Waveform-5000	+	+	+	–	+	–

The comparison in Table 11 clearly indicates that the proposed approach is generally more accurate than **DTP** (in the case of REPTree and SimpleCart). In the case of J48, each approach performs better on four datasets. The pruned standard DTs are smaller in REPTree and SimpleCart, but J48 gives the smallest DTs for $\mathcal{IUDT}$.

Table 12 compares the results of $\mathcal{IUDT}$ with those from the application of data sampling **IUDTSD**. It is clear that the accuracy of $\mathcal{IUDT}$ is much better than that of **IUDTSD**, but in contrast to the accuracy results, the size results show that the sampling approach has some advantages. In general, the table indicates that the accuracy results of $\mathcal{IUDT}$ are better when using the REPTree and SimpleCart models, but the DTs constructed are larger than the sampling approach results. In the case of J48, we have equality between the accuracy and size results. This shows that using data sampling certainly provides a less complex DT, albeit at the cost of robustness.

Table 13 compares the results of $\mathcal{IUDT}$ with **IUDTAS**, in which only attribute selection is applied. Clearly, the accuracy and size results from the proposed approach are better. In general, the table indicates that $\mathcal{IUDT}$ gives better accuracy and size results with the REPTree and SimpleCart models, except in the case of J48 DT sizes, which are larger than those of the attribute selection approach. These results show that using a combination of attribute selection and data sampling provides a robust and less complex DT.

6. Application to fault diagnosis in a rotating machine

We now consider the application of the proposed method to fault diagnosis in rotating machines. Some of the main faults affecting the proper functioning of such machines were produced experimentally on a test rig. The condition-monitoring task can be converted to a classification task, where each condition (good and defective) is considered as a class. The target is to extract information from vibration sensors to indicate the machine's current condition (class). The approaches investigated in this paper are then used to seek the non-complex construction of effective decision rules, following the schematic shown in Fig. 9.

6.1. Experimental study

The test rig shown in Fig. 10 is composed of three shafts, two gears (one with 60 teeth and the other with 48 teeth), six bearing housings, a coupling, and a toothed belt. The system is driven by a DC variable-speed electric motor with a rotational speed ranging from 0 to 1500 rpm.

Vibration signatures were used to monitor the condition of the test rig. The vibration signals were acquired using an accelerometer fixed on the bearing housing, connected to a data acquisition system equipped with OROS software. Vibration signatures were recorded under three different rotational speeds (300, 900, and 1500 rpm) under a normal operating condition and with three different faults: mass imbalance, gear fault, and faulty belt.

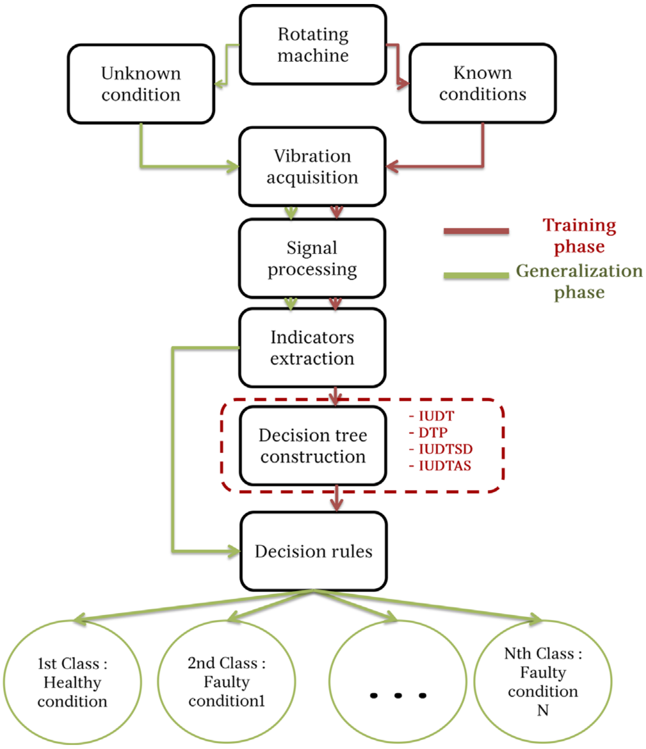

Fig. 9. Optimized DTC stages for fault diagnosis of rotating machines.

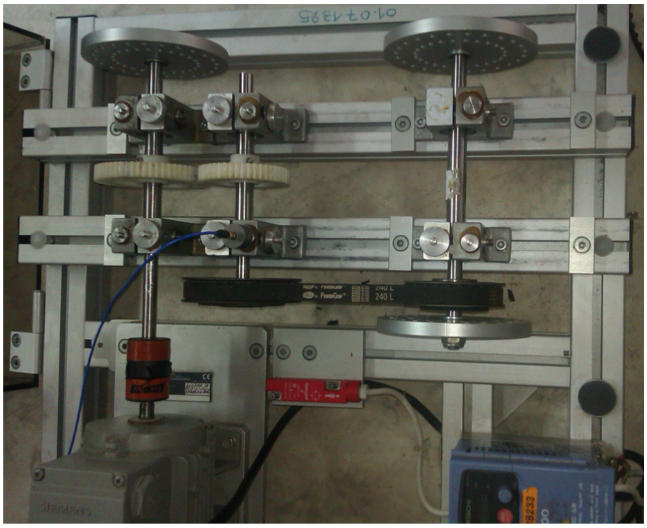

Fig. 10. Test rig (URASM-CSC Laboratory).

6.2. Signal processing

The wavelet transform has been widely studied over the past two decades, and its use has seen a significant growth and interest in vibration analysis. The formulation of its discrete variant (*DWT*), which requires less computation time than the continuous form, is shown in the following equation:

$$DWT(j,k)=\frac{1}{\sqrt{2^j}}\int_{-\infty}^{+\infty} s(t)\psi^*\left(\frac{t-2^jk}{2^j}\right)dt \tag{4}$$

Mallat (1989) introduced an effective use of the discrete wavelet transform by applying a succession of filters on several levels. The resulting signals are called approximation coefficients and detail coefficients. To overcome the down-sampling encountered throughout the decomposition, the coefficients are subjected to reconstruction filters to create new signals called approximations (A) and details (D). Fig. 11 illustrates the principle of DWT decomposition. In the present study, Daubechies wavelets with two levels of decomposition were used to extract the approximations and details of the original signals. The frequency space was transformed by applying a Fast Fourier Transform (FFT) to each original signal, as well as to each of the approximations, details, and coefficients, as shown in Fig. 12 for the example of mass imbalance under a rotational speed of 900 rpm.

6.3. Extraction of Indicators

Thirty-five original signals were recorded under four different operating conditions (classes) and three different rotational speeds, giving a total of 420 original signals. The crest factor, root mean square, skewness, and variance were extracted from the temporal forms of these signals.

From the frequency spectra, we derived the maximum amplitude, its frequency, the frequency of the second highest amplitude, the interval between the two highest amplitude frequencies, the

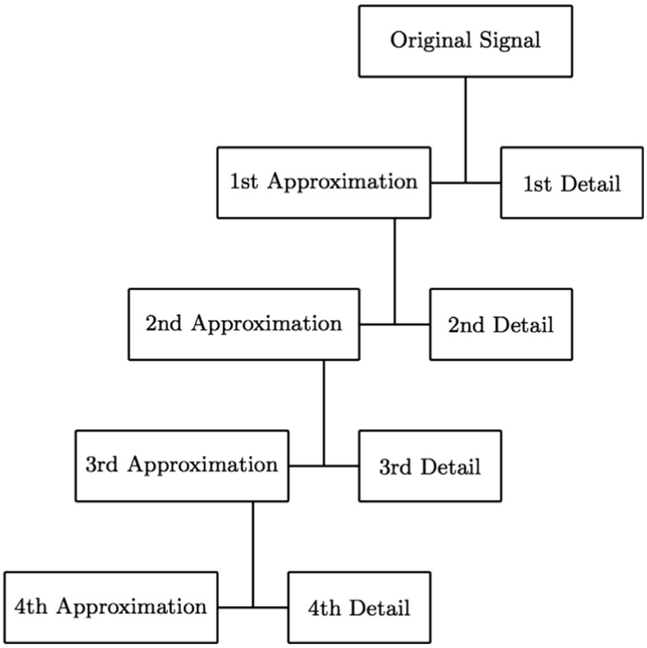

Fig. 11. DWT decomposition.

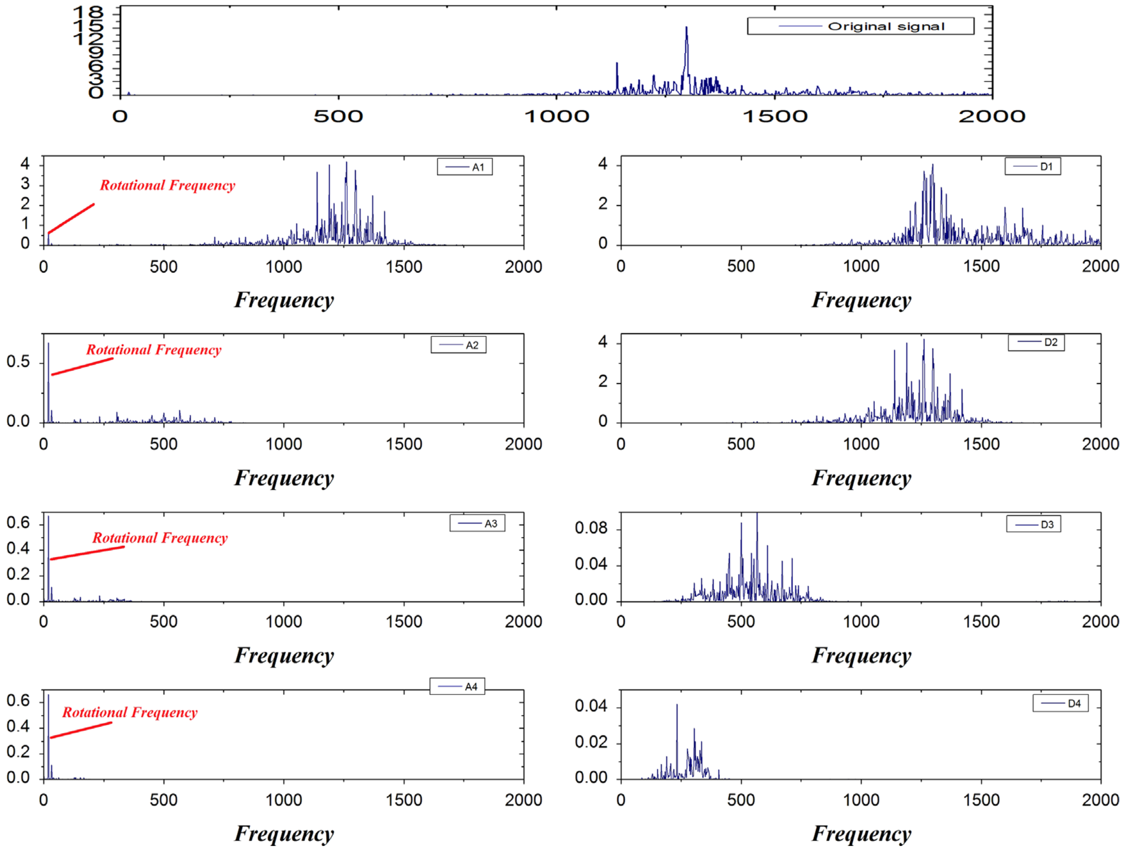

Fig. 12. Original signal, approximations, and detail spectra extracted from the test rig under the condition of mass imbalance.

Table 14
Fault diagnosis results for rotating machine application.

DTs	IUDT		DTP		IUDTSD		IUDTAS	
	Accuracy	Size	Accuracy	Size	Accuracy	Size	Accuracy	Size
J48	89.52	21	91.66	33	92.38	19	96.66	29
REPTree	98.41	21	85.31	27	92.14	19	90.47	29
SimpleCart	95.37	23	92.30	35	90.47	19	96.66	25

mean interval between the four highest amplitude frequencies, and the root mean square. These 11 indicators were also extracted from each of the four signals from the DWT decomposition application, giving a total of 55 indicators from the original signal.

6.4. Results and discussion

The experimental results using three DT algorithms are given in Table 14. These results indicate a global superiority in terms of classification accuracy. Slightly less complex trees are constructed with the **IUDTSD** approach, but these exhibit much lower performance. **IUDTSD** can also be seen to give an over-generalization, as explored in Section 3.2.

The best results were obtained using REPTree, and the decision rules produced by the experimental algorithms are given in the appendix. In terms of the *IUDT* output, *REPTree* gave the best classification accuracy. Table 15 lists the selected indicators used for its construction in appearance order in the tree (AOT), showing the signal from which each indicator originated.

From Table 15, we can see that only the indicators extracted from the original signal and the first-level decomposition were used to construct the best tree, where four decomposition levels were done. By exclusively extracting the retained indicators in the DTC, a significant saving in both storage memory and computation time can be achieved, particularly in diagnostic and maintenance tasks that require the archiving of data over long periods.

Table 15
Indicators used in the construction of the best tree.

AOT	Indicator	Originating signal
1	Skewness	Original signal
2	Root mean square of the spectra	Original signal
3	Signal root mean square	First approximation signal
4	Crest factor	First detail signal
5	Mean interval between the frequencies of the four maximum amplitudes	Original signal

7. Conclusion

The popularity of DTs is strongly related to their simplicity, ease of understanding, and close resemblance to human reasoning. However, each DT model has its own specific advantages and limitations, making the choice of a particular DT difficult to justify. The model choice depends strongly on the performance of the pruning approach, which forces users to select a model according to their requirements. This implies that DT users must study all growing and pruning techniques so as to choose the most appropriate model, which is a difficult task.

When constructing DTs, the pruning phase is useful for reducing the model's complexity and size, but often imparts a penalty on the accuracy, particularly for small datasets. In this paper, we proposed an improved algorithm without the pruning phase that combines the attribute selection and data sampling processes.

Experimental results on 10 benchmark datasets showed that the proposed approach improves the accuracy of DTs and effectively reduces their size, avoiding the problems of over- or under-pruning.

Finally, the $\mathcal{IUDT}$ approach was applied to the practical application of fault diagnosis in rotating equipment. This demonstrated its effectiveness in terms of classification accuracy and tree complexity. Moreover, extracting only indicators selected by the proposed approach allows a significant gain in storage memory and computation time.

We conclude that the proposed method is feasible, especially in the case of small datasets, and is an effective approach when exhaustive knowledge on data characteristics is missing.

Acknowledgments

In memoriam, we would like to gratefully acknowledge Prof. Lhouari Nourine and Prof. Engelbert Mephu Nguifo for their kind advice on this research.

Appendix A

```
REPTree DTP
============
sCF-Sigs  < 0.13
|   sCF-A2s  < 0.07
|   |   sfMXPx-Sigs  < 53.5 : 2.000000 (42/4) [25/7]
|   |   sfMXPx-Sigs  > = 53.5 : 1.000000 (7/0) [3/0]
|   sCF-A2s  > = 0.07
|   |   sMeanFFT-Sigs  < 0
|   |   |   sfMXPx-Sigs  < 52.5
|   |   |   |   sfMXPx-Sigs  < 10.5 : 1.000000 (18/0) [6/0]
|   |   |   |   sfMXPx-Sigs  > = 10.5
|   |   |   |   |   sfMXPx-Sigs  < 50.5 : 2.000000 (30/2) [12/0]
|   |   |   |   |   sfMXPx-Sigs  > = 50.5 : 1.000000 (18/2) [7/2]
|   |   |   sfMXPx- Sigs  > = 52.5
|   |   |   |   sCF-A2s  <  0.08 : 1.000000 (13/0) [6/2]
|   |   |   |   sCF-A2s  > = 0.08
|   |   |   |   |   sRMS-Sigs  <  3.03 : 1.000000 (2/0) [2/0]
|   |   |   |   |   sRMS-Sigs  > = 3.03 : 7.000000(25/0) [16/3]
|   |   sMeanFFT-Sigs  > = 0
|   |   |   sfMXPx-A1s  <  48 : 7.000000 (36/0) [17/0]
|   |   |   sfMXPx-A1s  > = 48
|   |   |   |   sfMXPx-Sigs  <  53 : 1.000000 (7/0) [4/0]
|   |   |   |   sfMXPx-Sigs  > = 53 : 7.000000(7/0) [3/0]
sCF-Sigs  > = 0.13
|   sRMS-Sigs  <  4.98
|   |   sRMS-Sigs  <  2.81 : 2.000000 (2/0) [1/0]
|   |   sRMS-Sigs  > = 2.81 : 1.000000 (3/2) [3/1]
|   sRMS-Sigs  > = 4.98 : 3.000000 (70/0) [35/0]
```

```
REPTree IUDTSD
================
sCF-Sigs  <  0.13
|   sfMXPx-Sigs  <  51
|   |   sRMS-Sigs  <  7
|   |   |   sRMS-Sigs  <  4.57 : 2.000000 (7/0) [0/0]
|   |   |   sRMS-Sigs  > = 4.57 : 1.000000 (7/0) [0/0]
|   |   sRMS-Sigs  > = 7
|   |   |   sCF-A2s  <  0.07 : 2.000000 (15/0) [0/0]
|   |   |   sCF-A2s  > = 0.07
|   |   |   |   sMeanFFT- Sigs  <  0 : 2.000000 (6/0) [0/0]
|   |   |   |   sMeanFFT-Sigs  > = 0 : 7.000000 (10/0) [0/0]
|   sfMXPx-Sigs  > = 51
|   |   sfMXPx-A2s  <  53.5 : 1.000000 (10/1) [0/0]
|   |   sfMXPx-A2s  > = 53.5
|   |   |   sMeanFFT-Sigs  <  0 : 1.000000 (6/0) [0/0]
|   |   |   sMeanFFT-Sigs  > = 0 : 7.000000 (16/0) [0/0]
sCF-Sigs  > = 0.13
|   sRMS-Sigs  <  5.11 : 2.000000 (2/1) [0/0]
|   sRMS-Sigs  > = 5.11 : 3.000000 (26/0) [0/0]
```

REPTree **IUDTAS**

```
===============
sMeanFFT-A2s < 0
|   sRMS-D2s < 1.28 : 2.000000 (15/0) [0/0]
|   sRMS-D2s > =1.28
|   |   sRMS-D2s < 3.94
|   |   |   sMeanFFT-A2s < 0
|   |   |   |   sRMS- D2s < 1.41
|   |   |   |   |   sMeanFFT-A2s < 0
|   |   |   |   |   |   sMeanFFT-A2s < 0 : 2.000000 (5/1) [0/0]
|   |   |   |   |   |   sMeanFFT-A2s > =0 : 1.000000 (4/0) [0/0]
|   |   |   |   |   sMeanFFT-A2s > =0 : 7.000000 (8/0) [0/0]
|   |   |   |   sRMS-D2s > =1.41
|   |   |   |   |   sMeanFFT-A2s < 0 : 1.000000 (21/0) [0/0]
|   |   |   |   |   sMeanFFT-A2s > =0
|   |   |   |   |   |   sMean-Freqdsit-A1s < 10.83 : 7.0000 (2/0) [0/0]
|   |   |   |   |   |   sMean-Freqdsit-A1s > =10.83 : 1.000 (8/0) [0/0]
|   |   |   sMeanFFT-A2s > =0 : 7.000000 (24/1) [0/0]
|   |   sRMS-D2s > =3.94
|   |   |   sMean-Freqdsit-A1s < 10.17 : 1.000000 (19/0) [0/0]
|   |   |   sMean-Freqdsit-A1s > =10.17 : 2.000000 (14/1) [0/0]
sMeanFFT-A2s > =0
|   sMeanFFT-A2s < 0.01
|   |   sCF-D1s < 0.14
|   |   |   sRMS-D2s < 15.12 : 7.000000 (17/0) [0/0]
|   |   |   sRMS- D2s > =15.12 : 3.000000 (2/0) [0/0]
|   |   sCF-D1s > =0.14
|   |   |   sCF-D1s < 0.17 : 3.000000 (10/1) [0/0]
|   |   |   sCF-D1s > =0.17 : 3.000000 (39/0) [0/0]

|   sMeanFFT-A2s > =0.01 : 2.000000 (22/0) [0/0]
```

REPTree *IUDT*

```
=============
sSkew-Sigs < 0.23
|   sMeanFFT-Sigs < 0.01
|   |   sMeanFFT-Sigs < 0
|   |   |   sRMS-A1s < 2.18 : 2.000000 (9/0) [0/0]
|   |   |   sRMS-A1s > =2.18
|   |   |   |   sRMS-A1s < 8.46
|   |   |   |   |   sMeanFFT-Sigs < 0
|   |   |   |   |   |   sMeanFFT- Sigs < 0 : 1.000000 (9/0) [0/0]
|   |   |   |   |   |   sMeanFFT-Sigs > =0
|   |   |   |   |   |   |   sMean-Freqdsit-Sigs < 11.83 : 7.00000 (7/0) [0/0]
|   |   |   |   |   |   |   sMean-Freqdsit-Sigs > =11.83 : 1.0000 (8/0) [0/0]
|   |   |   |   |   sMeanFFT-Sigs > =0 : 7.000000 (10/0) [0/0]
|   |   |   |   sRMS-A1s > =8.46
|   |   |   |   |   sMean-Freqdsit-Sigs < 9.67 : 1.000000 (8/0) [0/0]
|   |   |   |   |   sMean- Freqdsit-Sigs > =9.67 : 2.000000 (6/0) [0/0]
|   |   sMeanFFT-Sigs > =0
|   |   |   sCF-D1s < 0.15 : 7.000000 (11/0) [0/0]
|   |   |   sCF-D1s > =0.15 : 3.000000 (5/1) [0/0]
|   sMeanFFT-Sigs > =0.01 : 2.000000 (13/0) [0/0]
sSkew-Sigs > =0.23 : 3.000000 (19/0) [0/0]
```

References

Agrawal, R., Srikant, R., et al., 1994. Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pp. 487–499.

Badaoui, M.E., Guillet, F., Daniere, J., 2004. New applications of the real cepstrum to gear signals, including definition of a robust fault indicator. Mech. Syst. Signal Process. 18, 1031–1046.

Baydar, N., Ball, A., 2001. A comparative study of acoustic and vibration signals in detection of gear failures using Wigner–Ville distribution. Mech. Syst. Signal Process. 15, 1091–1107.

Bermejo, P., Gámez, J., Puerta, J., 2008. On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria. In: Proceedings of IPMU'08, pp. 638–645.

Bermejo, P., de la Ossa, L., Gámez, J.A., Puerta, J.M., 2012. Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. Knowl.-Based Syst. 25, 35–44.

Blake, C.L., Merz, C.J., 1998. UCI Repository of Machine Learning Databases 〈http://www.ics.uci.edu/~mlearn/mlrepository.html〉. University of California, Department of Information and Computer Science, Irvine, CA, p. 460.

Bourrillion, K., et al., 2010. Guava: Google Core Libraries for Java 1.5+.

Bradford, J.P., Kunz, C., Kohavi, R., Brunk, C., Brodley, C.E., 1998. Pruning decision trees with misclassification costs. In: Machine Learning, ECML-98. Springer, pp. 131–136.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth & Brooks, Monterey, CA.

Caruana, R., Freitag, D., 1994. How useful is relevance? FOCUS 14, 2.

Chen, C.S., Chen, J.S., 2011. Rotor fault diagnosis system based on SGA-based individual neural networks. Expert Syst. Appl. 38, 10822–10830.

Davey, B.A., 2002. Introduction to Lattices and Order. Cambridge University Press.

Deng, S., Lin, S.Y., Chang, W.L., 2011. Application of multiclass support vector machines for fault diagnosis of field air defense gun. Expert Syst. Appl. 38, 6007–6013.

Djebala, A., Ouelaa, N., Hamzaoui, N., 2008. Detection of rolling bearing defects using discrete wavelet analysis. Meccanica 43, 339–348.

Esposito, F., Malerba, D., Semeraro, G., Kay, J., 1997. A comparative analysis of methods for pruning decision trees. IEEE Trans. Pattern Anal. Mach. Intell. 19, 476–491.

Fournier, D., Crémilleux, B., 2002. A quality index for decision tree pruning. Knowl. Based Syst. 15, 37–43.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The WEKA data mining software: an update. ACM SIGKDD Explor. Newslett. 11, 10–18.

Hall, M.A., Holmes, G., 2003. Benchmarking attribute selection techniques for discrete class data mining. IEEE Trans. Knowl. Data Eng. 15, 1437–1447.

Huerta, E.B., Duval, B., Hao, J.K., 2006. A hybrid GA/SVM approach for gene selection and classification of microarray data. In: Applications of Evolutionary Computing. Springer, pp. 34–44.

Ishibuchi, H., Nakashima, T., Nii, M., 2001. Genetic-algorithm-based instance and feature selection. In: Instance Selection and Construction for Data Mining. Springer, pp. 95–112.

Kankar, P., Sharma, S.C., Harsha, S., 2011. Fault diagnosis of ball bearings using continuous wavelet transform. Appl. Soft Comput. 11, 2300–2312.

Karabadji, N.E.I., Seridi, H., Khelf, I., Laouar, L., 2012. Decision tree selection in an industrial machine fault diagnostics. In: Model and Data Engineering. Springer, pp. 129–140.

Khelf, I., Laouar, L., Bouchelaghem, A.M., Rémond, D., Saad, S., 2013. Adaptive fault diagnosis in rotating machines using indicators selection. Mech. Syst. Signal Process. 40, 452–468.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artif. Intell. 97, 273–324.

Liu, H., 2010. Instance Selection and Construction for Data Mining. Springer-Verlag.

Liu, H., Yu, L., 2005. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. Knowl. Data Eng. 17, 491–502.

Luo, L., Zhang, X., Peng, H., Lv, W., Zhang, Y., 2013. A new pruning method for decision tree based on structural risk of leaf node. Neural Comput. Appl., 1–10.

Macaš, M., Lhotská, L., Bakstein, E., Novák, D., Wild, J., Sieger, T., Vostatek, P., Jech, R., 2012. Wrapper feature selection for small sample size data driven by complete error estimates. Comput. Methods Prog. Biomed. 108, 138–150.

Mallat, S.G., 1989. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. 11, 674–693.

Mingers, J., 1987. Expert systems-rule induction with statistical data. J. Oper. Res. Soc., 39–47.

Mingers, J., 1989. An empirical comparison of pruning methods for decision tree induction. Mach. Learn. 4, 227–243.

Mosher, M., Pryor, A.H., Lewicki, D.G., 2003. Detailed vibration analysis of pinion gear with time–frequency methods. National Aeronautics and Space Administration, Ames Research Center.

Niblett, T., Bratko, I., 1987. Learning decision rules in noisy domains. In: Proceedings of Expert Systems' 86, The Sixth Annual Technical Conference on Research and Development in Expert Systems III. Cambridge University Press, pp. 25–34.

Oates, T., 1997. The effects of training set size on decision tree complexity. In: Proceedings of the 14th International Conference on Machine Learning. Morgan Kaufmann, pp. 254–262.

Piramuthu, S., 2008. Input data for decision trees. Expert Syst. Appl. 34, 1220–1226.

Provost, F., Domingos, P., 2003. Tree induction for probability-based ranking. Mach. Learn. 52, 199–215.

Quinlan, J.R., 1987. Simplifying decision trees. Int. J. Man–Mach. Stud. 27, 221–234.

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning, vol. 1. Morgan Kaufmann.

Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S., 2006. Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognit. 39, 2383–2392.

Sakthivel, N., Sugumaran, V., Babudevasenapati, S., 2010. Vibration based fault diagnosis of monoblock centrifugal pump using decision tree. Expert Syst. Appl. 37, 4040–4049.

Stracuzzi, D.J., Utgoff, P.E., 2004. Randomized variable elimination. J. Mach. Learn. Res. 5, 1331–1362.

Sugumaran, V., Muralidharan, V., Ramachandran, K., 2007. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. Mech. Syst. Signal Process. 21, 930–942.

Sugumaran, V., Ramachandran, K., 2007. Automatic rule learning using decision tree for fuzzy classifier in fault diagnosis of roller bearing. Mech. Syst. Signal Process. 21, 2237–2247.

Xing, E.P., Jordan, M.I., Karp, R.M., et al., 2001. Feature selection for high-dimensional genomic microarray data. In: ICML, pp. 601–608.

Yang, B.S., Lim, D.S., Tan, A.C.C., 2005. VIBEX: an expert system for vibration fault diagnosis of rotating machinery using decision tree and decision table. Expert Syst. Appl. 28, 735–742.

Yildiz, O.T., Alpaydin, E., 2005. Linear discriminant trees. Int. J. Pattern Recognit. Artif. Intell. 19, 323–353.

Zhao, Y., Zhang, Y., 2008. Comparison of decision tree methods for finding active objects. Adv. Space Res. 41, 1955–1959.