Master's Thesis

On Windowing as a sub-sampling method for Distributed Data Mining

David Martínez-Galicia

Director:
Dr. Alejandro Guerra Hernández

Co-directors:
Dr. Nicandro Cruz Ramírez
Dr. Héctor Xavier Limón Riaño

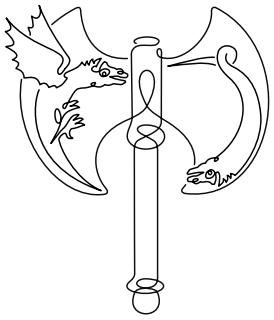
August 26, 2020

Maestría en Inteligencia Artificial
Centro de Investigación en Inteligencia Artificial
Universidad Veracruzana
Sebastián Camacho No 5, Xalapa, Veracruz
México, 91000.

Acknowledgements:
(Agradecimientos)

A mi madre y mi familia, por su cariño incondicional.
A mis amigos, por su cálida ayuda a lo largo de la maestría.
A cada uno de mis profesores, por sus enseñanzas dentro y fuera del salón.
A mis revisores y codirectores, por su retroalimentación para mejorar este trabajo.
Al Dr. Alejandro Guerra Hernández, por su guía para convertirme en un buen investigador.

“Solve et coagula”



Abstract

Windowing is a sub-sampling method, originally proposed to cope with large datasets when inducing decision trees with the ID3 and C4.5 algorithms. The method exhibits a strong negative correlation between the accuracy of the learned models and the number of examples used to induce them, i.e., the higher the accuracy of the obtained model, the fewer examples used to induce it. This paper contributes to a better understanding of this behavior in order to promote Windowing as a sub-sampling method for Distributed Data Mining. For this, the generalization of the behavior of Windowing beyond decision trees is established, by corroborating the observed negative correlation when adopting inductive algorithms of different nature. Then, focusing on decision trees, the windows (samples) and the obtained models are analyzed in terms of Minimum Description Length (MDL), Area Under the ROC Curve (AUC), Kulllback–Leibler divergence, and the similitude metric Sim1; and compared to those obtained when using traditional methods: random, balanced, and stratified samplings. It is shown that the aggressive sampling performed by Windowing, up to 3% of the original dataset, induces models that are significantly more accurate than those obtained from the traditional sampling methods, among which only the balanced sampling is comparable in terms of AUC. And finally, the study of the window evolution is analyzed observing the previous suggested metrics. Sim1, accuracy and AUC show an irregular increment through iterations, however in predictive terms the final model may not be the most accurate. Related to the MDL elements, results suggest that decisions trees grow in size to deal with unseen data, but at the same time their data compression capacity improves. Windowing also shows a behavior that favors more balanced distribution in the samples, but it is restricted by the number of class minority instances and their relevance. Although noisy domains difficult the mining of accurate models, Windowing drastically reduce the sample size, this behavior shows that Windowing is a competitive method in distributed scenarios. Results also suggest further experiments to enhance the performance of the method, i.e., Adopting metrics for detecting relevant, noisy, and redundant instances.

Contents

List of Figures

List of Tables

Introduction 1

Windowing is originally a procedure that enables the induction of Decision Trees, using a small sample of the training instances. The produced trees have comparable levels of accuracy with those obtained using the complete data, and also, are moderately less complex in terms of the number of nodes. These promising results and that Windowing automatically determines the sample size, encourage its adoption as a sub-sampling method for Distributed Data Mining (DDM). However, Windowing adoption is conditioned to its generalization beyond decision trees and the need of a deeper understanding of its behavior. This work looks for characterizing the performed sampling in terms of informational properties (related to redundancy, similarity, and class distribution), and studying the effect of data reduction on the predictive and data compression performance of the models.

In this chapter, the foundation of this thesis is presented, starting with the problem statement that highlights the main contribution of this work. Next, a justification of this study, briefly comparing with related work and remarking its novelty. In the following section, the main objectives that guides the research are presented. Then, the hypothesis, which proposes the outcome research, is stated. And finally, the related work to this dissertation and its contribution.

1.1 Problem statement

Even though Artificial Intelligence (AI) lacks a standardized definition, it has been characterized as the field of study that seeks to explain and emulate intelligent behavior in terms of computational processes [1]. Based on this definition, a computer should possess some capabilities in order to be described as having intelligent behavior. Machine learning, proposed by the Turing Test, is the ability of an entity to change its behavior in a way that improves its performance in the future [2]. This research addresses Machine Learning techniques as tools for the Knowledge Discovery in Databases (KDD).

Related to Machine Learning, Windowing is a technique, proposed by John Quinlan, to induce models from large datasets, those whose size precludes loading them in memory [3]. This method is

composed of two steps as shown in the algorithm 1. The first is the creation of a window, that is a small random sample of available instances in the training set, and the second step, that requires more computational resources, consists of inducing a model with the window and testing it on the remaining instances, such that all misclassified instances are moved to the window. This step iterates until a stop condition is reached, e.g., all the available examples are correctly classified or a desired level of accuracy is reached.

Algorithm 1: Windowing.

```
1  Function Windowing(Instances):
2      Window ← sample(Examples);
3      Examples ← Examples − Window ;
4      repeat
5          stopCond ← true ;
6          model ← induce(Window) ;
7          for example ∈ Examples do
8              if classify(model, example) ≠ class(example)
                then
9                  Window ← Window ∪ {example} ;
10                 Examples ← Examples − {example} ;
11                 stopCond ← false ;
12     until stopCond;
13     return Model ;
```

Although the lack of memory is not a problem nowadays, similar issues could arise when mining big distributed volumes of data. The motivation of this dissertation is founded in previous works [4–6] that suggest that Windowing offers a big data reduction to induce models with high performance, and this reduction is correlated with how difficult is to learn a problem. These previous discoveries impulse to study the properties of this method and the suffered transformations by the dataset to contribute to the field of DDM.

1.2 Hypothesis

Windowing exhibits consistent behavior through the use of different Machine Learning models in DDM scenarios, i.e., models with high levels of accuracy are induced from small samples. In these scenarios, it is possible to obtain gains in terms of performance, model complexity and data compression, against traditional subsampling methods.

1.3 Objectives

The main objective of this thesis is to study the behavior of Windowing through the use of different Machine Learning models in DDM scenarios. This study beholds the description of both, models and samples, to understand the performed sampling. Therefore, the specific research objectives are as follows:

1. Measuring the correlation between the accuracy of the learned model and the percentage of used instances for different datasets and different inductive learning algorithms.
2. Suggesting metrics that measure informational features (similarity, data compression, model complexity, etc.) to compare the windows, the original datasets and the induced models.
3. Comparing Windowing with other sub-sampling techniques to observe the advantages of its use.
4. Characterizing the operation of this technique on different types of datasets.
5. Providing a wide description about Windowing behavior and the best conditions to make use of it.

1.4 Justification

Related to the benefits of Windowing, Johannes Fürnkranz [6] has argued that this method offers three advantages:

1. It copes well with memory limitations, reducing considerably the number of examples required to induce a model of acceptable accuracy.
2. It offers an efficiency gain by reducing the time of convergence, especially when using a separate-and-conquer inductive algorithm, as Foil, instead of the divide-and-conquer algorithms like ID3 and C4.5.
3. It offers an accuracy gain, particularly in noiseless datasets, possibly explained by the fact that learning from a subset of examples may often result in a less over-fitting theory.

Articles related to JaCa-DDM [7, 8], a framework for Distributed Data Mining, that implements Windowing-based strategies, has shown:

1. A strong correlation between the accuracy of the learned Decision Trees and the percentage of examples used to induce them. That is, the higher the obtained accuracy, the fewer the used examples to induce the model.
2. The performed reductions are as big as more than 90% of the available training examples.

On the other hand, Jarryl Wirth et al. [9] published an early critic about the computational cost of Windowing and its inability to deal with noisy domains. Based on the previous benefits and observations, this dissertation extends the knowledge about Windowing in four directions:

1. It studies the generalization of this method as a sub-sampling mechanism in multiple Machine Learning algorithms.
2. It analyzes the windows and the models in terms of different informational properties.
3. It provides a comparison between Windowing and some of the state of the art sub-sampling algorithms.
4. It summarizes the best situations and conditions to use Windowing in DDM scenarios.

1.5 Related work

There has been a small amount of research on Windowing probably because the negative critics about its performance in noisy domains. The early work has focused on its definition, providing mechanisms to deal with the instances selection, such as the two versions proposed by John Quinlan [3]. Other early work has focused on the use of rule learning algorithms to improve the time convergence [4]. This section also surveys works that study its performance, suggest applications and propose enhancements for this method. The considered publications are ordered in chronological order.

Induction over large datasets

In teh work of John Quinlan [3] two versions of Windowing are introduced as techniques to deal with memory limitations to induce Decision trees.

- The first variant suggests a simple scheme where a Decision Tree is induced from a random sample of the available dataset called window, and then it is updated with the missclassified instances in the remaining examples.
- The second variant proposed a method where the window size remains fixed and the examples added are some key elements from the previous Decision Tree, missclassified instances, and random elements from the past window.

Even when both versions are based in the fact that it is possible to generate a correct model to explain a large collection of instances holding just a small part of this data in memory, this study just focused on the temporal complexity of the versions and not the

precision of the models. However,John Quinlan suggests especial attention to the initial size of the window and the limit of missclassified instances added in one iteration.

Learning efficient classification and their application to chess end game

John Quinlan [10] applies the first variant of Windowing (variable window size) to discover efficient classification procedures from large datasets in Chess domain. The experimentation results show that Windowing allows a fast convergence to accurate Decision Trees using a small fraction of the available dataset, but it do not offer a complex analysis of the behavior of this technique. It also states that Windowing performance is not very sensitive to parameters.

Experiments on the cost and benefits of Windowing in ID3

Jarryl Wirth et al. [9] formulate a strong critic to Windowing, referring that this method has a high computational cost. This study analyzes variables related to the temporal complexity of the method (the number of iterations and the CPU time), the window (the percentage of the training set used in the final iteration), and the model (the predictive accuracy of the final decision tree and its complexity measured by the number of nodes). Although Windowing obtained good results in metrics associated to the model and the window size, authors consider that Windowing may induce a final tree that do not capture the concepts as well as a tree built from the complete training set. This work also states that Windowing should be avoided in noisy domains because results suggest that the model complexity and the CPU time tend to increment.

Megainduction: a test flight

Jason Catlett [11] introduces a study case where the dataset is related to the space radiator subsystem in NASA's Space Shuttle. This work focuses in non-noisy scenarios and large datasets. Its main hypothesis is: if a small sample turns out to give results as good as the full set, induction on the full set is an unnecessary expense. It presents a full study of Windowing related to the time learning, the error rate and the models' size. The conclusion states that produced trees with Windowing were highly accurate, moderate small, and after being converted to rules, were judged by the expert to be not only comprehensible and acceptable, but to contain new knowledge that might otherwise have remained undiscovered. The conclusion proved with empirical evidence that Windowing is capable not only to find accurate patterns in the available data, but also new complex information, this contradicts the conclusion of

the previous work and positions Windowing as a tool capable of finding knowledge in large volumes of information.

Windowing in C4.5 Programs from Machine Learning

This book chapter [12], written by John R. Quinlan, justify the use of Windowing in three advantages: it sometimes leads to a faster construction of Decision Trees, it produces more accurate trees from a uniform-class initial window, and it allows the possible generation of multiple trees for a voting approach. It also proposed the following improvements: a uniform class distribution in the initial window, a control mechanism for the addition of examples, and a stop condition where the program can stop before it appears that the sequence of trees is not becoming more accurate. Experimentation in this thesis, unlike Quinlan's work [12], is realized with a stratified samples as first window to analyze the Windowing behavior, and the possible changes to the class distribution.

More efficient Windowing

Based in the studies and improvements of Windowing, Johannes Fürnkranz proposed the hypothesis that rule learning algorithms are more appropriate for Windowing than decision trees algorithms [4]. This paper presents a new enhancement for Windowing using a rule learning algorithm (DOS), and its conclusions state that for separate-and-conquer rule learning algorithms Windowing shows significant gains in efficiency and its performance is affected in noisy domains.

Noise-tolerant Windowing

Following his research line, Johannes Fürnkranz suggest a new improvement to tackle noisy domains [5]. This new enhancement is based in a noise-tolerant rule learning algorithm (RIP), and a new mechanism for the addition of new examples. This mechanism just considers examples that are covered by insignificant rules or positive examples that are not covered by any rule of the previous iteration.

Integrative Windowing

This paper [6] summarizes all the research of Fürnkranz proposing a final Windowing version. This version is a little far of its original formulation, but it considers the improvements of Fürnkranz' previous works like the use of the rule learning algorithm DOS and the removal of examples from the window if they are covered by consistent rules. The proposed version obtained comparable or better levels of accuracy than others, and it has an efficiency gain in time. Unlike Fürnkranz, this dissertation states that the advantages of Windowing as a sub-sampling method can be generalized beyond decision trees.

Modeling and implementing DDM strategies in JaCa-DDM

Xavier Limón [8] introduces a new framework for Distributed Data Mining using BDI agents. JaCa-DDM is a novel system founded on the agents and artifacts paradigm that was conceived to design, implement, deploy and evaluate learning strategies. The strategies developed in this software are grouped in four sets: centralized, centralizing, meta-learning, and Windowing-based strategies. The results in Windowing-based strategies suggest a negative correlation between the ratio of training instances used to obtain a model and its accuracy. In other words, if a good accuracy can be achieved, then the percentage of used training instances decreases, and vice versa. Possibly, this relation is due to consistent, redundant datasets.

Windowing strategy for Distributed Data Mining optimized through GPUs

Following this research line, Xavier Limón proposed two improvements for a new Windowing-based strategy in JaCa-DDM [7]. The first improvement exploits the fact that some counter examples seem redundant, if two counter examples reach the same tree leaf when classified, they are alike in the sense that they were misclassified for similar reasons. The second enhancement accelerates the search of counter examples using GPUs, this requires representing the decision tree and the training examples in data structures well suited for CUDA. The results shows a reduction of the used examples up to 95%, while preserving accuracy and a improve of specificity.

1.6 Contributions

This dissertation provides the next contributions:

1. The empirical evidence that the use of Windowing can be generalized to other Machine Learning algorithms.
2. A methodology that involves different Theory Information metrics to characterize the data transformation performed by a sampling.
3. The implementation of the proposed metrics available in a digital repository *, which contains:
   - Two java classes to calculate the Minimum Description Length and the Area Under the Curve [13, 14]. This classes use the Weka library and assume the use of decision trees as models.

* https://github.com/DMGalicia/Thesis-Windowing

- A python script to compute the following metrics: Sim1, Kullback-Leibler Divergence, Red, and a report of the dataset class distribution [15–17].

4. Two papers (shown in Appendix D) as result of our participation in the Mexican International Conference on Artificial Intelligence (MICAI).

Background 2

Before starting a detailed description of the methodological design, this chapter provides a short piece of background information on key concepts of KDD, Data Mining (DM), and Machine Learning (ML). It is not an exhaustive tutorial for these areas, but it is essential to understand the main terminology that supports this research. Furthermore, this chapter provides a review of some areas within the KDD process that share the same goals as Windowing.

2.1 KDD - Knowledge Discovery on Databases

Many people treat DM as a synonym for KDD because both fields have a data-driven approach, however Usama Fayyad et al. [18] define KDD as the overall process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. The basic problem addressed by the KDD process is one of mapping low-level data into other forms that might be more compact, abstract, or helpful.

The KDD process is interactive and iterative, involving nine steps with many decisions made by the user:

1. Identify the goal of the process, understand the application domain and gather prior knowledge.
2. Create a dataset focusing on a subset of variables or data samples, on which the discovery will be performed.
3. Clean and preprocess data, which include removing noise, and strategies for handling missing data fields.
4. Reduce the data to a appropriate format, for instance, removing not helpful variables.
5. Determine the goal of the KDD process, for example, summarization, classification, regression, clustering, etc.
6. Choose the data mining approach. This choice often depends on the collected data and the end user's preference.
7. Search for patterns of interest in a particular representational form such as classification rules or trees.
8. Interpret and visualize the mined patterns, or possibly return to any of the previous steps.
9. Use the interpreted results for further actions. This step also includes resolving potential conflicts.

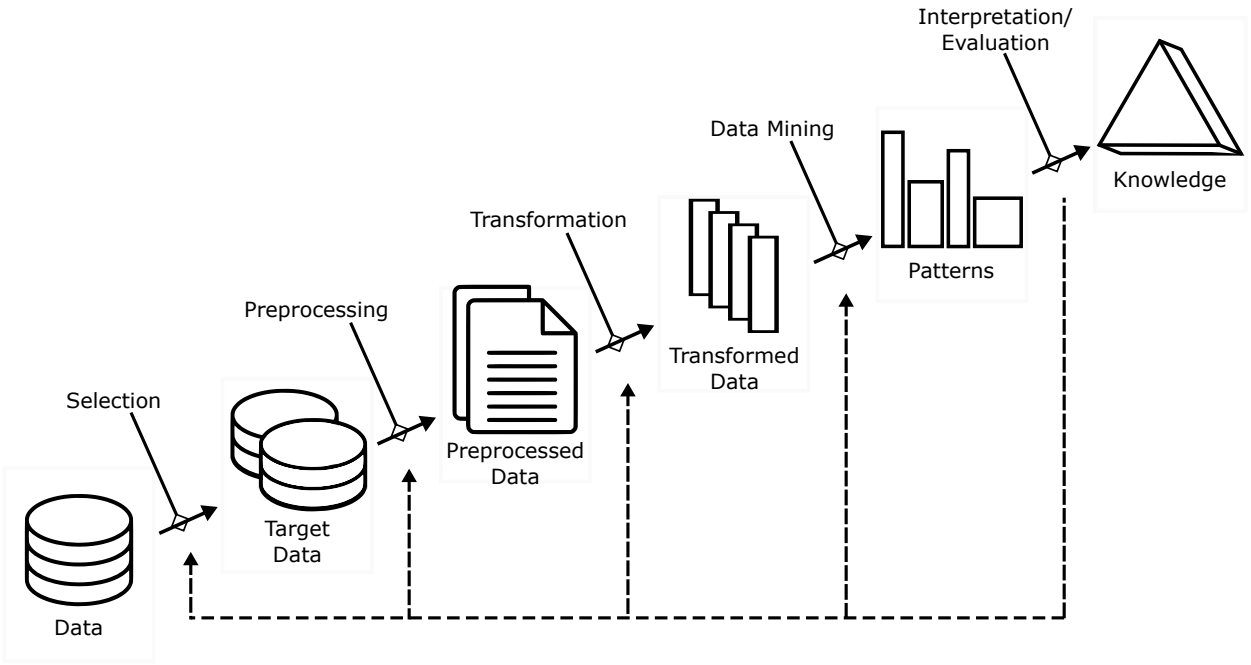


Figure 2.1: Steps that compose the KDD process.

Figure 2.1 shows the basic flow of steps in the KDD process, however, this can involve significant iterations and can contain loops. Because the interests of this dissertation, the following sections are focused on terminology related to step 7, the data mining. However, the other steps are as important for the successful application of KDD in practice.

2.2 Data Mining

DM is a step in the KDD process that consists of applying data analysis and discovery algorithms that produce a particular enumeration of patterns (or models) over the data [18]. It has two high-level primary goals:

1. **Prediction** involves using some variables or fields in the dataset to predict unknown or future values of other variable of interest.
2. **Description** focuses on finding human-understandable relations describing the data.

In context of the DM step, it is important to choose the correct approach to find valuables patterns in the data. This is often done using a high number of techniques from ML, pattern recognition, and statistics [19]. The field ML focuses on the development of algorithms that adapt the presentation of new information discovered in datasets, these techniques try to imitate the ability to

learn from the human being through experience and achieve an assigned task without external assistance [20].

ML methods are divided depending on the data format and the application purpose:

1. Classification methods try to find models that can describe data classes or concepts [21]. This work focuses on two special cases: binary classification where a data vector is classified into one, and only one, of two non-overlapping categories, and the multi-class classification where the number of non-overlapping categories grows [22].
2. Regression predicts missing or unavailable numerical data values rather than (discrete) class labels. It also encompasses the identification of distribution trends based on the available data [21].
3. Clustering is a descriptive task where an algorithm seeks to identify a finite set of categories (clusters) to describe the dataset. The cluster can be mutually exclusive and exhaustive or consist of a richer representation, such as hierarchical or overlapping categories [18, 23].
4. Summarization involves methods for finding a compact description for a subset of data. A simple example would be tabulating the mean and standard deviations for all the numeric variables [18].
5. Dependency modeling consists of finding a model that describes significant dependencies between variables. Dependency models exist at two levels: the structural level that specifies which variables are locally dependent on each other and the quantitative level that specifies the strengths of the dependencies using some numeric scale [18].
6. Change and deviation detection focuses on discovering the most significant changes in the data from previously measured or normative values [18].

Related to distributed scenarios where the data is placed in different location, Distributed Data Mining (DMM), is a special case that concerns the application of the traditional DM procedures trying to optimize the available resources such as communication costs, computing units and memory storage, in distributed environments [24].

2.3 Performance metrics

During the KDD process, supervised ML algorithms induce models from tagged data, that is, from data vectors that have a class assigned to them. However, this type of practice does not always

ensure good performance. Multiple performance metrics have been proposed to evaluate the classification results.

Performance measures for binary classification

Generally, the validation techniques provide a summary of the performance of each classifier using a test dataset. The summary is given in a table-like format known as a confusion matrix where the rows can represent the number of instances in a predicted class and the columns the number of instances in a real class (or vice versa) [21]. Figure 2.2 describes the composition of the confusion matrix, as well as some metrics that can be used in binary classification. By convention, within the binary classification the labels refer to positive cases and negative cases because problems, such as detection of a disease and approval of credits, are very common. However, this can apply to problems where classes are not contrary cases.

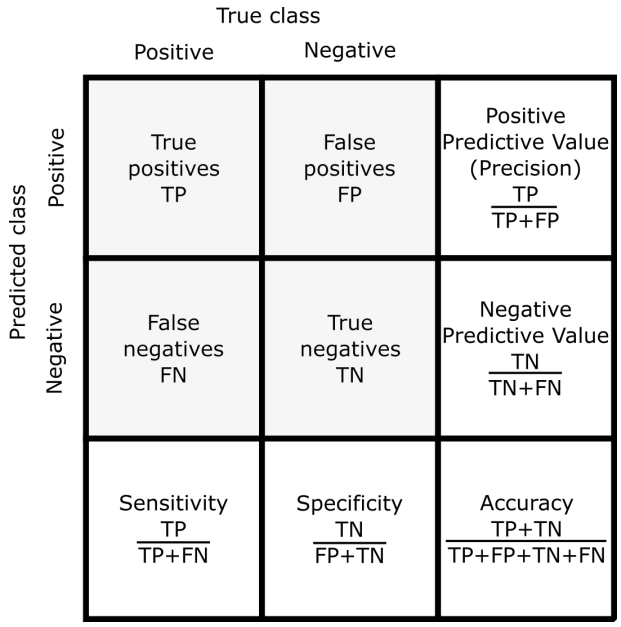


Figure 2.2: Confusion matrix.

- **Accuracy:** Also known as *recognition rate*. It is defined as the percentage of instances correctly classified [22, 25].

$$\frac{TP + VN}{TP + FP + VN + TN}$$

- **Positive predictive value (Precision):** It is defined as the percentage of correctly classified instances of all positive instances obtained by the classifier, in other words, it measures the precision of positive predictions [25].

$$\frac{TP}{TP + FP}$$

- **Negative predictive value:** It is defined as the percentage of correctly classified instances of all negative instances ob-

tained by classifier, in other words, it measures the precision of negative predictions [22].

$$\frac{TN}{TN + FN}$$

- **Sensitivity:** Also known as *recall*. It is defined as the proportion of correctly classified instances of the total number of positive observations, that is, the ability of a classifier to recognize positive instances [22, 25].

$$\frac{TP}{TP + FN}$$

- **Specificity:** It is defined as the proportion of correctly classified instances of the total number of negative observations, that is, the ability of a classifier to recognize negative instances [22].

$$\frac{TN}{FP + TN}$$

- **F-score:** It is defined as the measure of the precision of a test and requires *recall* and *precision* to calculate its value [22, 25].

$$2 \cdot \frac{precision \cdot recall}{precision + recall}$$

- **AUC:** Represents the probability that a random example is correctly classified. The AUC value ranges from 0 to 1. A model whose predictions are 100 % correct has an AUC of 1.0, if its predictions are not so correct, the AUC value tends to decrease. For its calculation, *sensitivity (recall)* and *specificity* are required [22, 26].

$$\frac{1}{2}\left(recall + specificity\right)$$

Performance measures for multi-class classification

Multi-class classification is the generalization of the binary case. Within this task, the evaluation measures are also calculated from the information obtained from a confusion matrix. These metrics can be calculated through two ways:

- **Micro average:** The values TP, FP, TN, FN are obtained by respectively adding the results for each class $i = 1, 2, \ldots, n$ using a one-against-all approach, and finally, the calculation of the metric is performed as in the binary case. This method is convenient when the classes are not balanced [27].
- **Macro Average:** The metrics for each class are independently calculated $i = 1, 2, \ldots, n$ (using a one-against-all approach), and at the end, their unweighted mean. This method is effective when the classes are balanced [27].

2.4 Approaches related to Windowing

Most DM algorithms from statistics, pattern recognition, and ML assume data are in the main memory and pay no attention to memory limitations, however Windowing was conceived to deal with this issue. Windowing behavior can be evaluated using two factors, the data reduction and the model performance. This section seeks to compare Windowing with techniques belonging to different stages of the KDD process.

The first factor is pretty close to goals of techniques related to the preprocessing stages, for example, sub-sampling an instance selection methods. These techniques are conceived under the hypothesis that a reduced sample can represent the nature of the original dataset.

Data sub-sampling is a statistical analysis procedure to select, manipulate, and analyze a representative subset of the training data. Although, these algorithms are characterized by not using heuristics and selecting the instances only once, they are highly employed in DM due to their low computational cost.

The selection of instances, in the case of reduction, is a problem that is related to two factors: the sample size and the precision in the classification [28]. According to Huan Liu, instance selection techniques can be seen as an optimization problem because it involves minimizing a set of instances and maximizing the performance of a model [29].

Instance selection methods can be classified by their evaluation method:

1. **Wrapper:** These techniques share an iterative approach as Windowing. Their selection criteria are based on the precision obtained by a classifier (commonly those instances that do not contribute to the classification precision are discarded) [30].
2. **Filter:** These methods use several selection criteria, but none are based on a model's feedback of the data [31]. Work related to filter methods selects the instances based on principles of clustering or on some features of the examples. This type of techniques is further from the operation of Windowing.

On the other hand, techniques from Ensemble learning and Active Learning focused on the same factors that Windowing, but their performance is ruled by different type of mechanisms and heuristics.

Ensemble learning is a ML paradigm in which multiple models are trained to solve the same problem. Unlike traditional ML

approaches that attempt to learn a hypothesis directly from training data, Ensemble learning methods attempt to build a set of models from partitions of the data and combine them to use [32].

Windowing-based algorithms are also closely related to the field of Active Learning. According to its definition coined within the ML area, this branch includes any type of learning in which the algorithm has some control over the data with which it is trained [33]. Techniques from Active Learning presuppose a scenario with desirable characteristics such as:

1. An initial labeled training set.
2. An oracle that provides the correct kind of instances.
3. A continuous flow, a pool of data or a classifier that generates instances.

2.5 State of the art

This section presents a review of the state-of-the-art of Windowing related techniques. During the KDD process, it is necessary to adopt data preprocessing techniques for preparing unstructured data and getting high-quality DM results. Windowing tackles the data preprocessing using an approach similar to the traditional sub-sampling methods. It selects a minor part of the training examples, while it tries to induce models with significant levels of accuracy.

The previous section listed four different approaches used by KDD techniques that share the same goals as Windowing. Regarding methods that select a sub-sample of the examples, the first two subsections describe the current methods used in the sub-sampling and instance selection areas. Concerning techniques that aim to improve the model performance selecting a part of the training examples, the last two subsections review the algorithms used in Ensemble Learning and Active Learning.

Statistical Sub-sampling

The statistical sub-sampling is a well-known solution to the apparent intractability of learning from datasets of large size. Unlike Windowing, these techniques are not iterative, i.e., they generate just one sample. Traditional sub-sampling techniques include:

- **Random sampling:** Select a subset of random examples. Sample size is a parameter to set [34].
- **Duplicate compaction:** Remove the repeated instances of the training data [34].

- **Stratified Sampling:** This technique is applied when the class values are not evenly distributed in order to obtain a sample with a class distribution similar to the original dataset' [34].

Instance Selection

There are a number of related studies suggesting Instance Selection methods for obtaining better performance in DM tasks. Specifically, Jankowski and Grochowski [35] surveyed several relevant selection techniques. The methods considered in this section are wrappers, because their performance are closer to Windowing's:

- **Condensed Nearest Neighbor Rule (CNN):** The algorithm starts with a new dataset N containing one instance per class. Instances are chosen at random from the training set. After that, each misclassified instance of the training set is moved to N. This procedure is very fragile regarding noise and presentation order [36].
- **Reduced Nearest Neighbor (RNN):** This algorithm use the dasaset result of the CNN algorithm. Then, it iterates over the examples and deletes only those examples that do not decrease accuracy [37].
- **IB3:** This algorithm has an incremental approach, where a instance x from the training set is added to a new set S, if the nearest acceptable instance in S (if there are no acceptable instance a random one is used) has different class than x [38]. IB3 employs a significance test to determine which instances are acceptable.
- **Encoding length - ELH:** This algorithm uses the cost function defined by:

  $$J(m,n,x) = F(m,n)+m*log_2(c)+F(x,n-m)+x*log_2(c-1)$$

  where n and m are instance numbers in the training set and in the new data S respectively. x defines the number of misclassified instances (based on S) and $F(m,n)$ is defined by:

  $$F(m,n) = log^*\left(\sum_{i=0}^{m}\frac{n!}{i!(n-i)!}\right)$$

  $log^*n = argmin_k F(k) \geq n$, k - is an integer, $F(0) = 1$ and $F(i) = 2^{F(i-1)}$. ELH starts with an empty set and adds instances only if they minimize the cost function $J(.)$ [39].

Ensemble learning

Like Windowing, Ensemble Learning techniques focus on samples, however, the samples are no constructed in many iterations. This review just consider general methods, i.e., those than can use any ML model.

- **Bagging:** It generates m new training sets of the same size, sampling uniformly and with replacement. Then the m models are fitted using the previous bootstrap samples and combined by averaging the result (for regression) or voting (for classification) [40]. This is meant to provide optimal coverage of the domain space, making more robust the predictive performance. Figure 2.3 shows a schematic illustration of how bagging works.

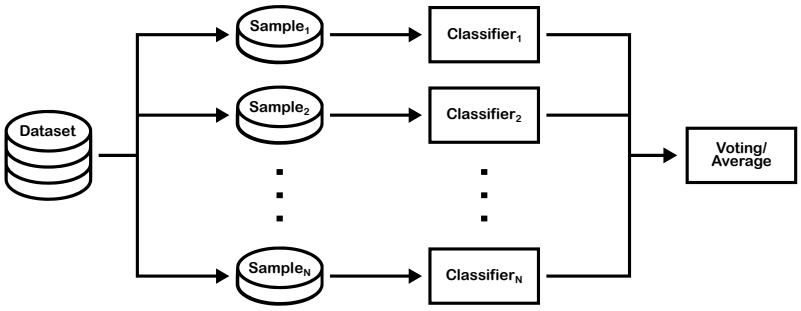


Figure 2.3: Bagging.

- **Boosting:** It trains weak classifiers sequentially, each trying to correct its predecessor, as depicted in Figure 2.4. In this technique, initially, the instances are equally weighted. After each iteration of the algorithm, the instances that are correctly classified are weighted lower than the incorrectly classified instances [41].

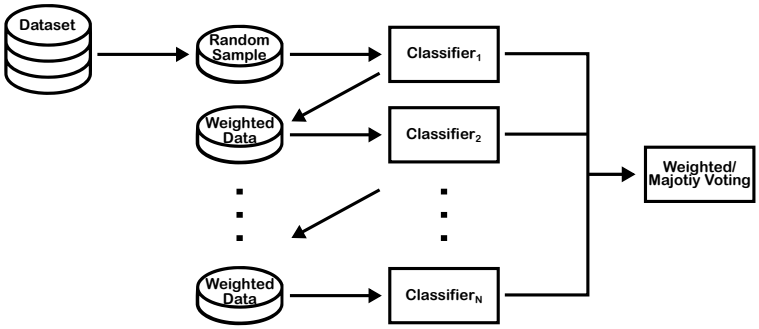


Figure 2.4: Boosting.

- **Stacking:** (Also known as stacked generalization) This approach uses a meta-learning algorithm to learn how to best

combine the predictions from two or more base ML algorithms of different nature. The meta-model is fitted with the predictions made by base models on a test set [42]. The outputs from the base models used as input to the meta-model may be real value in the case of regression, and class labels in the case of classification. Figure 2.5 shows how stacking works.

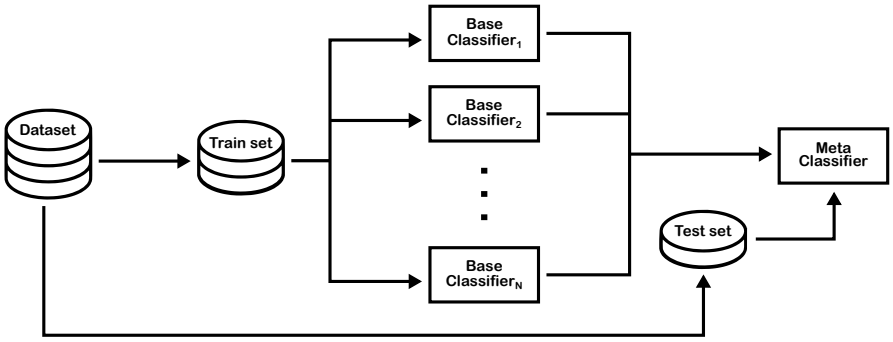


Figure 2.5: Stacking.

Other works as partitioning [43, 44]. focused on a specific type of ML model. However these techniques are not reviewed because proving the generalization of Windowing is one of the main objectives of this dissertation.

Active Learning

Active Learning formally studies the closed-loop phenomenon of a learner selecting actions or making queries that influence what data are added to its training set. This characteristic is similar to the construction of the windows in Windowing. However, Active Learning presupposed a continuous flow data. The following list review some of the most popular techniques in this paradigm:

- **Uncertainty sampling:** This technique assumes that there is a small set of data tagged L and a large set of unlabeled data U available. Typically, all instances in U are evaluated, or if U is very large, a sub-sample of it. Once evaluated, the instances that were classified with less confidence will be added to the training set in the next iteration, after consulting the oracle for their true classes [45]. Figure 2.6 shows the workflow of the uncertainty sampling.
  This approach has two main different strategies to determine uncertainty:

  1. Least Confidence Sampling: selects the instance whose most confident prediction is the least likely among the unlabeled instances available for querying [46].

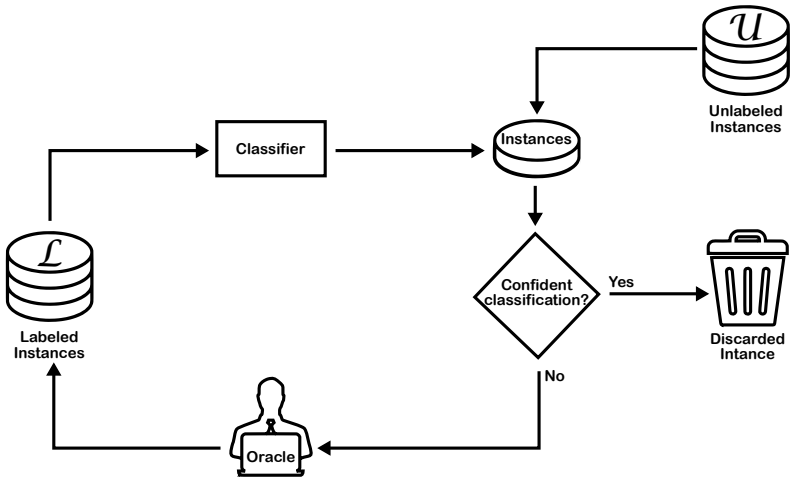


Figure 2.6: Uncertainty Sampling.

2. Margin of Confidence Sampling: selects the instances that minimizes the difference between the top two most confident predictions [47].

In definition, this technique is very similar to Windowing, except that the data pool is labeled and, therefore, it is possible to do the process without the oracle.

- **Selective sampling:** In this configuration (Figure 2.7), it is assumed that obtaining an unclassified instance is free or cheap. Based on this assumption, it then selects each unlabeled instance one at a time and allows the classifier to determine whether it wants to query the instance tag with the oracle or reject it based on its information [33]. To determine the relevance of an instance, it can use an informational metric as the entropy [48].

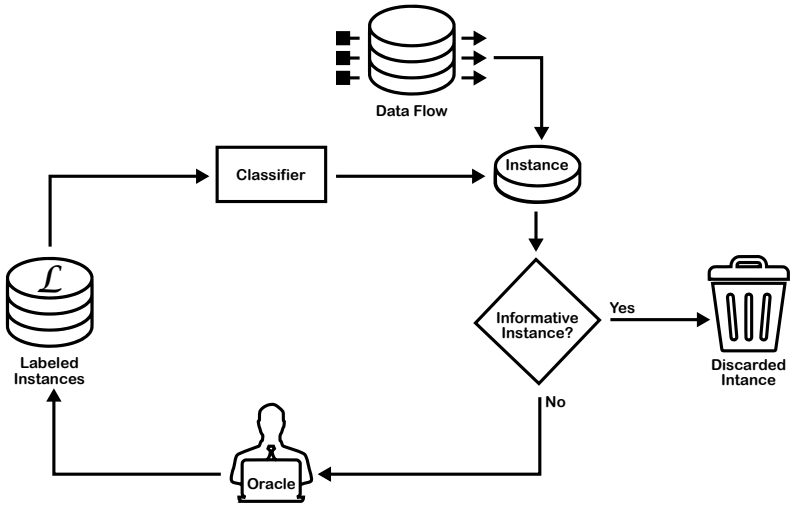


Figure 2.7: Selective Sampling.

- **Query by committee algorithm:** As the selective sampling algorithm, it is a selective method, the fundamental difference between them is that the query by committee algorithm has a multi-classifier approach, as shown in Figure 2.8. In the original conception, several hypotheses are taken at random from the version space [49]. The obtained committee is used to examine the unlabeled dataset. If there is a disagreement between the hypotheses regarding the class of an instance, this instance will be classified by the oracle.

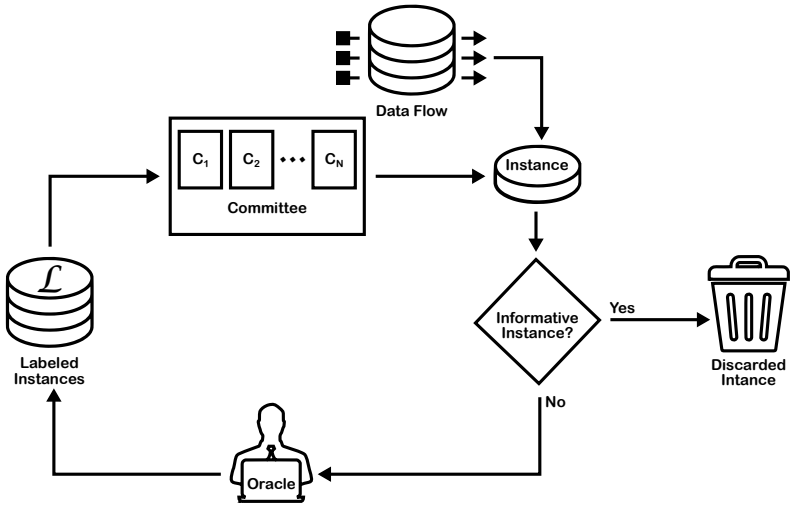


Figure 2.8: Query by committee algorithm.

- **Committee-based sampling:** This technique is based on the Query by committee algorithm, adapting its use for probabilistic classifiers [48].

Methodology 3

This chapter presents the most relevant methods and includes a description of the system JaCa-DDM, the planned experiments, theoretic measures to evaluate the results and the description of the statistical tests that is employed in order to analyze the results.

3.1 JaCa-DDM

Because this thesis is focused in distributed settings, JaCa-DDM * was adopted to run experiments. This tool is a DDM system founded on the Agents and Artifacts paradigm, conceived to design, implement, deploy, and evaluate learning strategies [8]. JaCa-DDM also provides a model which is a guideline to implement strategies. This model is built on the concepts of strategy and its deployment.

JaCa-DDM strategy: It is a 4-tuple $\langle Ags, Arts, Params, ag_1 \rangle$ that defines a workflow in terms of the involved agents and artifacts.

- $Ags = \{ag_1, \ldots, ag_n\}$ is the set of user-defined agent programs.
- $Arts = \{art_1, \ldots, art_m\}$ is the set of user-defined artifact types.
- $Params = \{param_1 : type_1, \ldots, param_k : type_k, \}$ is a set of parameters and their associated data types, where $type_{1,\ldots,k} \in \{int, bool, double, string\}$.
- $ag_1 \in Ags$ is a special agent program that acts as intermediary between the deployment system and the rest of the agents.

JaCa-DDM deployment: It is a 6-tuple $\langle Nodes, DS, Arts, Strat, Config, ag_0 \rangle$ that deals with configuration, distribution, and evaluation issues.

- $Nodes = \{node_0, node_1 \ldots, node_j\}$ is a set of computational nodes, usually distributed in a network, where: $node_0$ is running Jason and CArtAgO, while $node_{1,\ldots,j}$ are running only CArtAgO. Each node is denoted by a pair $\langle nodeName, IPaddress : port \rangle$.

* https://github.com/xl666/jaca-ddm

- $DS = \{ds_1, \ldots, ds_j\}$ is a set of data sources associated with each node, excepting $node_0$.
- $Arts = \{art_1, \ldots, art_i\}$ is a set of primitive artifact types, used to deploy the system.
- $Strat$ is a learning strategy as stated in **JaCa-DDM startegy**.
- $Config = \langle \delta, \pi \rangle$ is a configuration for a strategy deployment. It has two components:
  - $\delta = \{(ag, node, i), \ldots\}$ is a specification of how many copies of a given agent program will be focusing on a given node, where $ag \in Strat_{Ags}$ is an agent program in the strategy that will be copy $i \geq 1$ times, and assigned to focus on $node \in Nodes$.
  - $\pi = \{(p, v), \ldots\}$ is a set of pairs strategy parameter and initialization value.
- ag_0 is an agent program that deploys and configures the system.

The way the agents learn together using their artifacts, is implemented in the agent programs, while the deployment is defined in an XML description.

3.2 Counter strategy

JaCa-DDM defines a set of Windowing-based strategies using J48, the Weka [2] implementation of C4.5, as inductive algorithm. Due to the great similarity of the counter strategy to Windowing's original formulation, it was selected for our experimentation. This strategy consists in gathering all the counterexamples (missclassified instances) found in a node and sending them to the classifier artifact used to build the learned model, for updating it. The process continues until a stop condition is met [8]. However, it is important to remark the principal differences:

1. The dataset may be distributed in different sites, instead of the traditional approach based on a single dataset in a single site.
2. The loop for collecting the misclassified examples to be added to the window is performed by a set of agents using copies of the model distributed among the available sites, in a round-robin fashion.
3. The initial window is a stratified sample, instead of a random one.
4. An auto-adjustable stop criteria is combined with a configurable maximum number of iterations.

The components of this strategy are as follows:

- $Ags = \{contactPerson, worker, roundController\}$, where:
  - *contactPerson* controls the rounds and induces the learned model, beyond its basic competencies.
  - *worker* gathers counterexamples. Every worker is focused on a single node.
  - *roundController* determines if the auto-adjust stop condition has been met.
- $Arts = \{Classifier, InstancesBase, Evaluator\}$, where:
  - $Classifier$ is used to induce models and classify instances.
  - $InstancesBase$ is used to store and manipulate the learning examples.
  - $Evaluator$ is used to compute the accuracy of a model given a validation set. Used by the auto-adjust stop procedure.
- $Params = \{Classifier : String,$
  $Pruning : Bool,$
  $InitPercentage : Double,$
  $ValidationPercForRounds : Double,$
  $ChangeStep : Double,$
  $MaxRounds : Int\}$, where:
  - $Classifier$ specifies the adopted learning algorithm.
  - $Prunning$ forces the learning algorithms to use post-pruning if it is true.
  - $InitPercentage$ defines the size of the initial training set, i.e., the initial window size.
  - $ValidationPercForRounds$ defines the size of the validation set for the auto-adjust stop procedure.
  - $ChangeStep$ defines a threshold of minimum change between two consecutive rounds. Used by the auto-adjusted stop procedure.
  - $MaxRounds$ defines the maximum number of rounds.

A typical configuration for deploying the counter strategy is:

- $\delta = \{(contactPerson, node_1, 1), (roundController, node_1, 1), (worker, node_1, 1), \ldots, (worker, node_j, 1)\}$;
- $\pi = \{(Classifier, J48), (Pruning, true), (InitPercentage, 0.25), (ValidationPercForRounds, 0.20), (ChangeStep, 0.35), (MaxRounds, 15)\}$.

3.3 Datasets

The experiments are tested on the datasets shown in Table 3.1, selected from the UCI and MOA [50, 51] repositories. They vary in

the number of instances, attributes, and class' values; as well as in the type of the attributes. Some of them are affected by missing values. Literature [7] reports experiments on larger datasets, up to 4.8×10^6 instances, exploiting GPUs. However, datasets with higher dimensions are problematic, e.g., imdb-D with 1002 attributes does not converge using the Counter strategy.

Table 3.1: Datasets, adopted from UCI and MOA.

Dataset	Instances	Attribs	Types	Missing	Class
Adult	48842	15	Mixed	Yes	2
Australian	690	15	Mixed	No	2
Breast	683	10	Numeric	No	2
Diabetes	768	9	Mixed	No	2
Ecoli	336	8	Numeric	No	8
German	1000	21	Mixed	No	2
Hypothyroid	3772	30	Mixed	Yes	4
Kr-vs-kp	3196	37	Numeric	No	2
Letter	20000	17	Mixed	No	26
Mushroom	8124	23	Nominal	Yes	2
Poker-lsn	829201	11	Mixed	No	10
Segment	2310	20	Numeric	No	7
Sick	3772	30	Mixed	Yes	2
Splice	3190	61	Nominal	No	3
Waveform5000	5000	41	Numeric	No	3

3.4 Experiment A: On Windowing generalization

Experiment A seeks two objectives: the first one is to corroborate the correlation between the accuracy of the learned model and the percentage of used instances for the datasets in Table 3.1. Whereas, the second objective is to provide practical evidence about the suitable generalization of Windowing. For this, different Weka classifiers are adopted that replace J48. JaCa-DDM allows easy replacement and configuration of the new classifier artifacts of the system, namely:

Naive Bayes. A probabilistic classifier based on Bayes' theorem with a strong assumption of independence among attributes [52].

jRip. An inductive rule learner based on RIPPER that builds a set of rules while minimizing the amount of error [53].

Multilayer-perceptron. A multi-layer perceptron trained by back-propagation with sigmoid nodes except for numeric classes, in which case the output nodes become unthresholded linear units [54].

SMO. An implementation of John Platt's sequential minimal optimization algorithm for training a support vector classifier [55].

All classifiers are induced by running a 10-fold stratified cross-validation on each dataset, then observing the average accuracy of the obtained models and the average percentage of the original dataset used to induce the model, i.e., 100% means the full original dataset was used to create the window. In order to explain some features about the performed sub-sampling, the next measures are used to compare the final windows and the original dataset:

1. The model accuracy defined as the percentage of correctly classified instances.

$$100 \cdot \left(\frac{TP + TN}{TP + FP + TN + FN} \right)$$

where TP, TN, FP and FN respectively stand for the true positive, true negative, false positive, and false negative classifications using the test data. This measure ranges in value from 0 to 100, where 100 means a perfect predictive performance.

2. The Kullback-Leibler divergence (D_{KL}) [16] is defined as:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \cdot log_2 \left(\frac{P(x)}{Q(x)} \right)$$

where P and Q are probability distributions for the full dataset and the window, both are defined on the same probability space X, and x represents a class in the distribution. Instead of using a model to represent a conditional distribution of variables, as usual, we focus on the class distribution, computed as the marginal probability. Values closer to zero reflect higher similarity.

3. Sim_1 [15] is a similarity measure between datasets defined as:

$$sim_1(D_i, D_j) = \frac{|Item(D_i) \cap Item(D_j)|}{|Item(D_i) \cup Item(D_j)|}$$

where D_i is the window and D_j is the full dataset; and $Item(D)$ denotes the set of pairs attribute-value occurring in D. Values closer to one reflect higher similarity.

4. Red [17] measures redundancy in a dataset in terms of conditional population entropy (CPE), defined as:

$$CPE = -\sum_{i=1}^{n_c} p(c_i) \sum_{a=1}^{n_a} \sum_{v=1}^{n_{v_a}} p(x_{a,v}|c_i) \cdot log_2 p(x_{a,v}|c_i)$$

where n_c is the number of classes, n_a is the number of attributes, and n_{v_a} is the number of values for the attribute a. c_i stands for the $i-th$ class and $x_{a,v}$ represents the $v-th$ value of attribute a. CPE can be normalized [6] in such a way

that values closer to zero reflect lower redundancy:

$$Red = 1 - \frac{CPE}{\sum_{a=1}^{n_a} log_2 n_{v_a}}$$

For this experiment, 8 distributed sites were simulated on a machine, using the original implementation of the counter strategy adopting the configuration suggested by Xavier Limón [8] (Table 3.2).

Parameter	Value
Maximum number of rounds	10
Initial percentage for the window	0.20
Validation percentage for the test	0.25
Change step of accuracy every round	0.35

Table 3.2: Parameter configuration used in the Counter strategy (Experiment A).

The *change step* parameter defines a threshold. If the accuracy of the current model compared with the accuracy of the previous model surpasses this parameter, then other round is computed, otherwise, the process stops.

3.5 Experiment B: Properties of samples and models obtained by Windowing

The second experiment pursues a deeper understanding of the informational properties of the computed models, as well as those of the samples obtained by Windowing, i.e., the final windows. For this, given the positive results of the first experiment, we focus exclusively on decision trees (J48), for which new metrics to evaluate performance, complexity and data compression are well known. They include:

1. The Area Under the ROC Curve (AUC) defined as the probability of a random instance to be correctly classified. This measure ranges in value from 0 to 100, where 100 means a perfect predictive performance [22].

$$AUC = \frac{1}{2} \cdot \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) \cdot 100 \qquad (3.1)$$

   Even though AUC was conceived for binary classification problems, Foster Provost [14] proposes an implementation for multi-class problems based in the weighted average of AUC metrics for every class using a one-against-all approach, and the weight for every AUC is calculated as the class' appearance frequency in the data $p(c_i)$.

$$AUC_{total} = \sum_{c_i \in C} AUC(c_i) \cdot p(c_i)$$

2. The Minimum Description Length (MDL) principle states that the best model to infer from a dataset is the one which minimizes the sum of the length of the model $L(H)$, and the length of the data when encoded using the theory as a predictor for the data $L(D|H)$ [56].

$$MDL = L(H) + L(D|H)$$

For decision trees, John Quinlan [13] proposes the next definition:

a) The number of bits needed to encode a tree is:

$$L(H) = n_{nodes} * (1 + ln(n_{attributes})) + n_{leaves}(1 + ln(n_{classes}))$$

Where n_{nodes}, $n_{attributes}$, n_{leaves} and $n_{classes}$ stand for the number of nodes, attributes, leaves and classes. This encoding uses a recursive top-down, depth-first procedure, where a tree which is not a leaf is encoded by a sequence of 1, the attribute code at his root, and the respective encodings of the subtrees. If a tree or subtree is a leaf, its enconding is a sequence of 0, and the class code.

b) The number of bits needed to encode the data using the decision tree is:

$$L(D|H) = \sum_{l \in Leaves} log_2(b+1) + log_2\left(\binom{n}{k}\right)$$

where n is the number of instances, k is the number of positives instances for binary classification and b is a known a priori upper bound on k, typically $b = n$. For non-binary classification, Quinlan proposes a iterative approach where exceptions are sorted by their frequency, and then codified with the previous formula. Models with less number of bits to encode $L(H)$ and $L(D|H)$ are preferred since they are simpler and smaller.

These metrics are used to compare the sample (the window) and the model computed by Windowing, against those obtained as follows, once a random sample of the original data set is reserved as test set:

- Without sampling, using all the available data to induce the model.
- By Random sampling, where any instance has the same selection probability [57].
- By Stratified random sampling, where the instances are subdivided by their class into subgroups, the number of

selected instances per subgroup is defined as the division of the sample size by the number of instances [57].

- By Balanced random sampling, as stratified random sampling, the instances are subdivided by their class into subgroups, but the number of selected instances per subgroup is defined as the division of the sample size by the number of subgroups, this allows the same number of instances per class [57].

Ten repetitions of 10-fold stratified cross-validation are run on each dataset, using the configuration suggested by Xavier Limón [8] (Table 3.3). The maximum number of rounds are increased to analyze the Windowing behavior without restrictions.

Parameter	**Value**
Maximum number of rounds	15
Initial percentage for the window	0.20
Validation percentage for the test	0.25
Change step of accuracy every round	0.35

Table 3.3: Parameter configuration used in the Counter strategy (Experiments B and C).

For a fair comparison, all the samples have the size of the window being compared. Statistical validity of the results is established following the method proposed by Demšar [58]. This approach enables the comparison of multiple algorithms on multiple data sets. It is based on the use of the Friedman test with a corresponding post-hoc test. Let R_i^j be the rank of the j^{th} of k algorithms on the i^{th} of N data sets. The Friedman test [59, 60] compares the average ranks of algorithms, $R_j = \frac{1}{N}\sum_i R_i^j$. Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks R_j should be equal, the Friedman statistic:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]$$

is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and k are big enough ($N > 10$ and $k > 5$). For a smaller number of algorithms and data sets, exact critical values have been computed [61]. Iman and Davenport [62] showed that Friedman's χ_F^2 is undesirably conservative and derived an adjusted statistic:

$$F_f = \frac{(N-1)\times\chi_F^2}{N\times(k-1)-\chi_F^2}$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. If the null hypothesis of similar performances is rejected, then the Nemenyi post-hoc test is realized

for pairwise comparisons. The performance of two classifiers is significantly different if their corresponding average ranks differ by at least the critical difference:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$.

When multiple classifiers are compared, the results of the post-hoc tests can be visually represented with a Critical Distance (CD) diagram. This compact, information-dense visualization consists on a main axis where the average rank of each methods is plotted along with a line that represents the Critical Difference (CD). Methods separated by a distance shorter than the CD are statistically indistinguishable, i.e., the evidence is not sufficient to conclude whether they have a similar performance and are connected by a black line. Figure 3.1 shows an example of two groups of algorithms, elements within each group do not have significant differences in performance. In contrast, methods separated by a distance larger than the CD have a statistically significant difference in performance. The best performing methods are those with lower rank values shown on the left of the diagram.

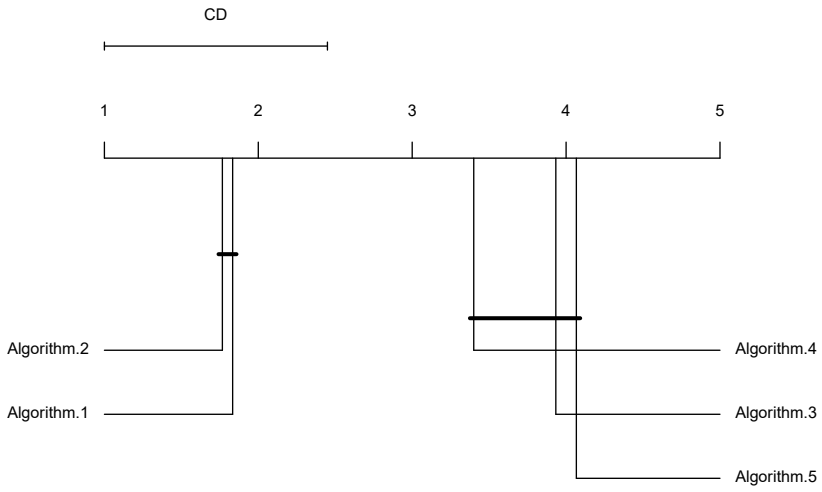


Figure 3.1: Example of a CD diagram.

3.6 Experiment C: Window evolution over time

Experiment C aims to yield a full description about the evolution of windows and their effects on the model. For this, the counter strategy was slightly modify in order to save the first windows and the resulting windows every iteration. A 10-fold stratified

cross-validation is run by every dataset, observing the average of the explained metrics in Section 3.4 (Experiment A: On Windowing generalization) and Section 3.5 (Experiment B: Properties of samples and models obtained by Windowing). This experiment adopts the setting in Table 3.3, 8 simulated nodes and Decision Trees as classifiers.

3.7 Computer specifications

Because this dissertation just seeks the characterization of sampling performed by Windowing and its effects in the induction, the simulation of 8 computational nodes is run in one machine with the following specifications:

Table 3.4: Computer specifications.

Specification	**Value**
Processor	Intel Core i5-8300H
Speed processor	2.30 GHz
RAM	8.00 GB

Results 4

4.1 Experiment A

Figure 4.1 shows a strong negative correlation between the percentage of training instances used to induce the models and their accuracy, independently of the adopted inductive algorithm. This reproduces the results for J48 reported in literature [8] and corroborates that under Windowing, in general, the models with higher accuracy require less examples to be induced.

However, accuracy is affected by the adopted inductive algorithm, e.g., Poker-lsn is approached very well by J48 (99.75 ± 0.07 of accuracy) requiring few examples (5% of the full dataset); while

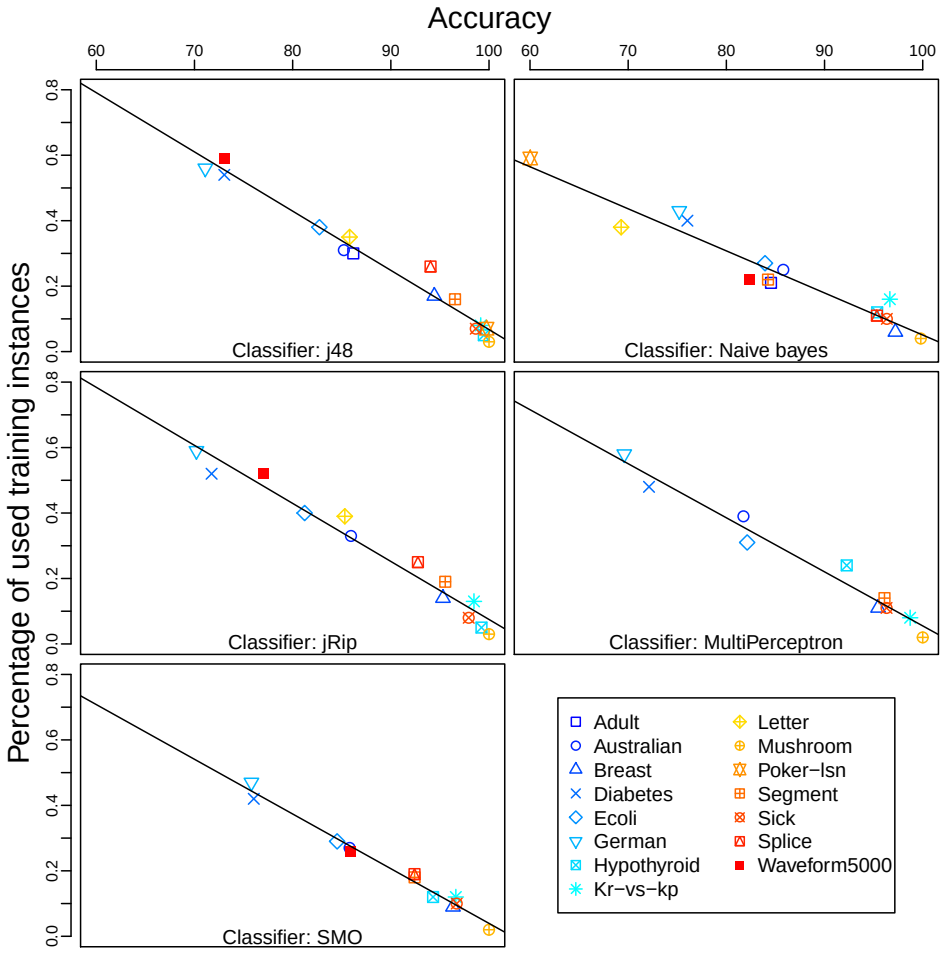


Figure 4.1: Correlation between accuracy and percentage of used training examples.
J48 = -0.98, NB = -0.96, jRip = -0.98, MP = -0.98 and SMO = -0.99.

Naive Bayes is not quite successful in this case (60.02 ± 0.42 of accuracy) requiring more examples (59%). This behavior is also observed between jRip and MultiPerceptron for Hypothyroid; and between SMO and jRip for Waveform5000. This is possibly due to the properties of the model that the different algorithms induce. A decision tree is not the same as a discriminator based on Naive Bayes, and so on. Not all algorithms finish on reasonable time because the temporal complexity of each one depends on the type of classifier induced and the features of the dataset.

Table 4.1 shows the accuracy results in detail, where accuracies are comparable to those obtained without using Windowing, i.e., using 100% of the available data for induction. Big datasets, as Adult, Letter, Poker-Isn, Splice, and Waveform5000 did not finish on reasonable time when using jRip, MultiPerceptron and SMO, with and without Windowing. In such cases, results are reported as not available (na). This might be solved by running the experiments in a real cluster of 8 nodes, instead of simulating them.

Table 4.1: Accuracies obtained from 10-fold cross validation (wW = with Windowing, woW= Windowing, na = not available).

	J48	**NB**	**jRip**	**MP**	**SMO**
Adult (woW)	85.98 ± 0.28	83.24 ± 0.19	na	na	na
Adult (wW)	86.17 ± 0.55	84.54 ± 0.62	na	na	na
Australian (woW)	87.10 ± 0.65	85.45 ± 1.57	84.44 ± 1.78	83.10 ± 1.28	86.71 ± 1.43
Australian (wW)	85.21 ± 4.77	85.79 ± 4.25	85.94 ± 3.93	81.74 ± 6.31	85.80 ± 4.77
Breast (woW)	96.16 ± 0.38	97.84 ± 0.51	95.03 ± 0.89	96.84 ± 0.77	96.67 ± 0.40
Breast (wW)	94.42 ± 3.97	97.21 ± 2.34	95.31 ± 2.75	95.45 ± 3.14	96.33 ± 3.12
Diabetes (woW)	72.95 ± 0.77	75.83 ± 1.17	78.27 ± 1.81	74.51 ± 1.46	78.02 ± 1.79
Diabetes (wW)	73.03 ± 3.99	76.03 ± 4.33	71.74 ± 7.67	72.12 ± 4.00	76.04 ± 3.51
Ecoli (woW)	84.44 ± 1.32	83.50 ± 1.64	82.25 ± 3.11	83.69 ± 1.44	83.93 ± 1.31
Ecoli (wW)	82.72 ± 6.81	83.93 ± 7.00	81.22 ± 6.63	82.12 ± 7.49	84.53 ± 4.11
German (woW)	73.89 ± 1.59	76.94 ± 2.29	70.06 ± 0.90	70.26 ± 0.96	74.55 ± 1.76
German (wW)	71.10 ± 5.40	75.20 ± 2.82	70.20 ± 3.85	69.60 ± 4.84	75.80 ± 3.12
Hypothyroid (woW)	99.48 ± 0.20	95.72 ± 0.68	99.60 ± 0.15	94.38 ± 0.25	94.01 ± 0.48
Hypothyroid (wW)	99.46 ± 0.17	95.36 ± 0.99	99.23 ± 0.48	92.26 ± 2.75	94.30 ± 0.53
Kr-vs-kp (woW)	99.31 ± 0.06	87.68 ± 0.43	99.37 ± 0.29	99.06 ± 0.13	96.67 ± 0.37
Kr-vs-kp (wW)	99.15 ± 0.66	96.65 ± 0.84	98.46 ± 0.95	98.72 ± 0.54	96.62 ± 0.75
Letter (woW)	87.81 ± 0.10	64.33 ± 0.28	86.34 ± 0.22	na	na
Letter (wW)	85.79 ± 1.24	69.28 ± 1.26	85.31 ± 1.06	na	na
Mushroom (woW)	100.0 ± 0.00	95.9 ± 0.32	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
Mushroom (wW)	100.0 ± 0.00	99.80 ± 0.16	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
Poker-lsn (woW)	99.79 ± 0.00	59.33 ± 0.03	na	na	na
Poker-lsn (wW)	99.75 ± 0.07	60.02 ± 0.42	na	na	na
Segment (woW)	96.02 ± 0.29	79.95 ± 0.69	95.25 ± 0.52	95.61 ± 0.91	92.97 ± 0.36
Segment (wW)	96.53 ± 1.47	84.24 ± 1.91	95.54 ± 1.55	96.10 ± 1.15	92.42 ± 1.87
Sick (woW)	98.88 ± 0.29	93.13 ± 0.43	98.19 ± 0.22	95.81 ± 0.45	93.70 ± 0.56
Sick (wW)	98.64 ± 0.53	96.34 ± 1.44	97.93 ± 0.95	96.32 ± 1.04	96.71 ± 0.77
Splice (woW)	93.81 ± 0.39	95.05 ± 0.36	94.19 ± 0.27	na	93.46 ± 0.48
Splice (wW)	94.04 ± 0.79	95.32 ± 1.07	92.75 ± 2.11	na	92.41 ± 1.34
Waveform5000 (woW)	75.58 ± 0.37	80.25 ± 0.33	79.54 ± 0.37	na	86.81 ± 0.21
Waveform5000 (wW)	73.06 ± 2.55	82.36 ± 1.64	77.02 ± 1.59	na	85.94 ± 1.32

Table 4.2 shows the number of used examples results, in terms of the percentage of the full dataset used for each inductive algorithm. One advantage of Windowing over other sub-sampling techniques is that its heuristic process defines the size of the final window. Based in the results, the data reduction ranges roughly from 40 to 95%.

Table 4.2: Percentage of the full dataset used for induction in Windowing (na = not available).

	J48	NB	jRip	MP	SMO
Adult	0.30 ± 0.01	0.21 ± 0.00	na	na	na
Australian	0.31 ± 0.02	0.25 ± 0.01	0.33 ± 0.02	0.39 ± 0.04	0.27 ± 0.01
Breast	0.17 ± 0.01	0.06 ± 0.00	0.14 ± 0.01	0.11 ± 0.01	0.09 ± 0.01
Diabetes	0.54 ± 0.05	0.40 ± 0.02	0.52 ± 0.04	0.48 ± 0.03	0.42 ± 0.02
Ecoli	0.38 ± 0.03	0.27 ± 0.01	0.40 ± 0.03	0.31 ± 0.03	0.29 ± 0.02
German	0.56 ± 0.04	0.43 ± 0.01	0.59 ± 0.02	0.58 ± 0.02	0.47 ± 0.02
Hypothyroid	0.05 ± 0.00	0.12 ± 0.01	0.05 ± 0.00	0.24 ± 0.01	0.12 ± 0.01
Kr-vs-kp	0.08 ± 0.01	0.16 ± 0.01	0.13 ± 0.00	0.08 ± 0.00	0.12 ± 0.00
Letter	0.35 ± 0.02	0.38 ± 0.00	0.39 ± 0.01	na	na
Mushroom	0.03 ± 0.00	0.04 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
Poker-lsn	0.05 ± 0.00	0.59 ± 0.00	na	na	na
Segment	0.16 ± 0.01	0.22 ± 0.01	0.19 ± 0.01	0.14 ± 0.01	0.18 ± 0.00
Sick	0.07 ± 0.00	0.10 ± 0.01	0.08 ± 0.00	0.11 ± 0.01	0.10 ± 0.00
Splice	0.26 ± 0.01	0.11 ± 0.00	0.25 ± 0.01	na	0.19 ± 0.00
Waveform5000	0.59 ± 0.02	0.22 ± 0.01	0.52 ± 0.00	na	0.26 ± 0.01

The KL divergence coefficient (Table 4.3) between the windows and the full datasets was close to zero in the most of the cases, evidencing that the class distribution of the windows is very similar to that observed in the full datasets. However, it does not seem to be a correlation between this coefficient and the accuracy, e.g., Mushroom has zero divergence and 100% of accuracy, but Waveform5000 has similar divergence but considerable lower accuracy.

Table 4.3: Kullback-Leibler divergence between the windows and the full datasets (na = not available).

	J48	NB	jRip	MP	SMO
Adult	0.10 ± 0.00	0.37 ± 0.01	na	na	na
Australian	0.02 ± 0.01	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Breast	0.05 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.06 ± 0.01	0.04 ± 0.01
Diabetes	0.02 ± 0.01	0.04 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.04 ± 0.00
Ecoli	0.11 ± 0.02	0.19 ± 0.02	0.12 ± 0.03	0.16 ± 0.04	0.23 ± 0.03
German	0.03 ± 0.01	0.06 ± 0.00	0.03 ± 0.00	0.01 ± 0.00	0.03 ± 0.00
Hypothyroid	0.24 ± 0.03	0.32 ± 0.05	0.25 ± 0.04	0.00 ± 0.00	0.52 ± 0.06
Kr-vs-kp	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00
Letter	0.02 ± 0.00	0.05 ± 0.00	0.03 ± 0.00	na	na
Mushroom	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Poker-lsn	0.19 ± 0.00	0.01 ± 0.00	na	na	na
Segment	0.24 ± 0.03	0.50 ± 0.03	0.22 ± 0.03	0.26 ± 0.02	0.31 ± 0.02
Sick	0.22 ± 0.04	0.20 ± 0.03	0.26 ± 0.04	0.24 ± 0.02	0.48 ± 0.01
Splice	0.03 ± 0.00	0.02 ± 0.01	0.02 ± 0.00	na	0.02 ± 0.00
Waveform5000	0.00 ± 0.00	0.15 ± 0.01	0.00 ± 0.00	na	0.00 ± 0.00

Table 4.4 shows the results for sim_1, suggesting that the windows for Australian, Breast, German, Letter, Kr-vs-Kp, and Poker-lsn conserve all the values for their attributes observed in the full datasets; while Adult and Segment have problems achieving this. As in the previous case, this notion of similarity neither seems to correlate with the observed accuracy, e.g., Segment.

Table 4.4: Table of similarity measure sim_1 (na = not available).

	j48	NB	jRip	MP	SMO
Adult	0.39±0.01	0.29±0.00	na	na	na
Australian	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Breast	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Diabetes	0.73±0.04	0.63±0.02	0.72 ± 0.03	0.69 ± 0.02	0.64 ± 0.01
Ecoli	0.77±0.03	0.65±0.02	0.78 ± 0.02	0.69 ± 0.04	0.65 ± 0.03
German	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00
Hypothyroid	0.45±0.01	1.00±0.01	0.48 ± 0.01	0.68 ± 0.01	0.59 ± 0.01
Kr-vs-kp	1.00±0.01	0.97±0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
Letter	0.99±0.01	0.99±0.01	0.98 ± 0.00	na	na
Mushroom	0.97±0.02	0.99±0.01	0.98 ± 0.00	0.97 ± 0.01	0.97 ± 0.01
Poker-lsn	1.00±0.00	1.00±0.00	na	na	na
Segment	0.28±0.01	0.32±0.01	0.31 ± 0.01	0.25 ± 0.01	0.28 ± 0.00
Sick	0.57±0.02	0.58±0.01	0.59 ± 0.01	0.60 ± 0.02	0.60 ± 0.01
Splice	0.97±0.04	0.96±0.05	0.97 ± 0.03	na	0.96 ± 0.04
Waveform5000	0.93±0.01	0.71±0.01	0.90 ± 0.00	na	0.76 ± 0.01

Red shows consistently the same values for the windows and the full datasets, meaning that both of them have very similar levels of redundancy. Given the nature of Windowing this can be a little bit surprising, since the window is expected to be less redundant than the full dataset because it does not include examples already covered by the induced models. But Red measures the information value given the information about the class values, an

Table 4.5: Table of redundancy measure using the 10-folds cross-validation windows (na = not available).

	Full	J48	NB	jRip	MP	SMO
Adult	0.71	0.72 ± 0.00	0.72 ± 0.00	na	na	na
Australian	0.63	0.61 ± 0.00	0.61 ± 0.01	0.61 ± 0.00	0.61 ± 0.00	0.61 ± 0.00
Breast	0.74	0.60 ± 0.01	0.58 ± 0.01	0.59 ± 0.01	0.59 ± 0.00	0.58 ± 0.01
Diabetes	0.58	0.58 ± 0.00	0.58 ± 0.00	0.58 ± 0.00	0.58 ± 0.00	0.58 ± 0.00
Ecoli	0.91	0.92 ± 0.00	0.92 ± 0.01	0.91 ± 0.00	0.92 ± 0.00	0.92 ± 0.00
German	0.62	0.62 ± 0.00	0.62 ± 0.00	0.61 ± 0.00	0.61 ± 0.00	0.61 ± 0.00
Hypothyroid	0.84	0.84 ± 0.00	0.84 ± 0.00	0.84 ± 0.00	0.84 ± 0.00	0.84 ± 0.00
Kr-vs-kp	0.72	0.72 ± 0.01	0.72 ± 0.00	0.70 ± 0.00	0.71 ± 0.00	0.71 ± 0.00
Letter	0.98	0.97 ± 0.00	0.97 ± 0.00	0.97 ± 0.00	na	na
Mushroom	0.71	0.69 ± 0.00	0.68 ± 0.00	0.68 ± 0.00	0.69 ± 0.00	0.69 ± 0.00
Poker-lsn	0.91	0.91 ± 0.00	0.91 ± 0.00	na	na	na
Segment	0.89	0.90 ± 0.00	0.89 ± 0.00	0.90 ± 0.00	0.90 ± 0.00	0.89 ± 0.00
Sick	0.68	0.67 ± 0.01	0.68 ± 0.00	0.67 ± 0.00	0.67 ± 0.00	0.67 ± 0.00
Splice	0.72	0.71 ± 0.01	0.71 ± 0.01	0.71 ± 0.00	na	0.70 ± 0.00
Waveform5000	0.69	0.69 ± 0.00	0.70 ± 0.00	0.69 ± 0.00	na	0.70 ± 0.00

intrinsic property of the data set; while the redundancy reduction expected by Windowing is a property of a dataset given a classifier, i.e., a dataset is redundant if when eliminating a random example, the induced model does not change. This behavior of Red, reported in literature [6], suggests that a different measure for redundancy should be adopted. Tables with the full information of the experiment A are showed in Appendix A.

4.2 Experiment B

For this experiment, the Poker-lsn dataset was excluded because the cross-validations runs do not finish on a reasonable time, and the metric Red was not adopted because it does not provide the information about redundancy we expected. Tables on Appendix B shows all the information related about the experiments and the metrics. The label Cross-Validation stands for a simple 10-folds cross-validation and it is the one that use the 90% of instances to induce a model. This technique was selected to observe the possible advantages and disadvantages of the use of most instances. In order to summarize the results, this chapter includes scatter plots of the former metrics results. Each plot does a comparison between the suggested techniques, and includes the average results (points) and their standard deviation (bars). A dotted blue line is also plotted, representing Windowing results of a given metric.

According to Kullback-Leibler Divergence (Figure 4.2), Windowing is the method that skews more the original class distribution in non-balanced datasets. Balanced sampling also shows high divergence, probably because it tends to uniform the class distribution and some datasets have unbalanced distributions, the divergence of this technique grows. The Experiment C (on Section 4.3) studies the windows evolution and its effect in metrics like the class distribution. The random sampling, the cross-validation and the stratified sampling, on the other hand, do not tend to modify the distribution.

Full-Dataset is, without surprise, the sample that gathers more attribute/values pairs from the original data, since it uses 90% of the available data. It is included in the results exclusively for comparison with the rest of the sampling methods. Figure 4.3 also shows that Windowing tends to collect more information content in most of the datasets compared with all the sampling, this is probably result of the heuristic nature of Windowing.

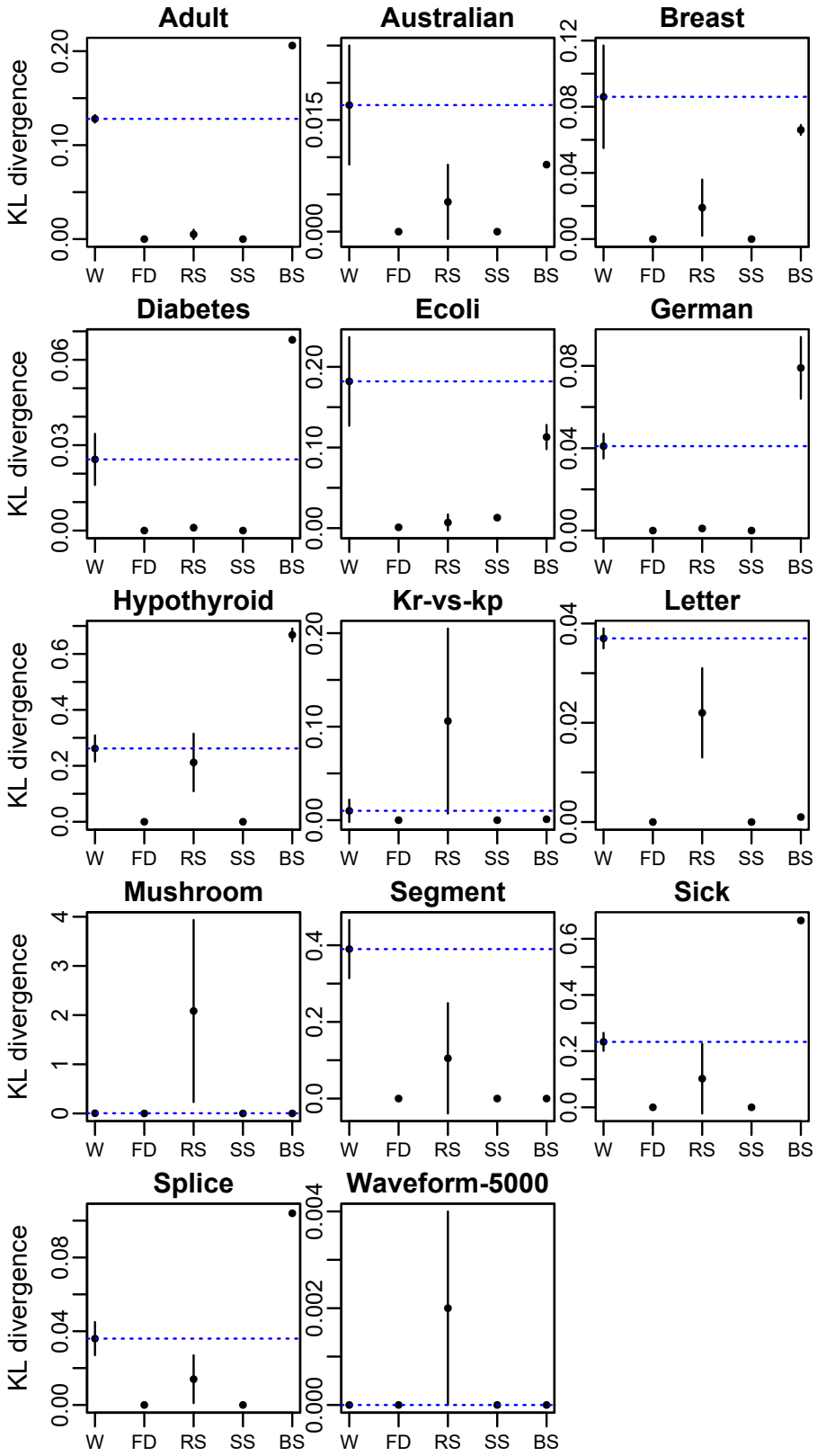


Figure 4.2: Average results of KL divergence.
W = Windowing, FD = Full-Dataset, RS = Random-Sampling, SS = Stratified-Sampling, and BS = Balanced-Sampling.
Values closer to zero reflect higher similarity with the original class distribution.

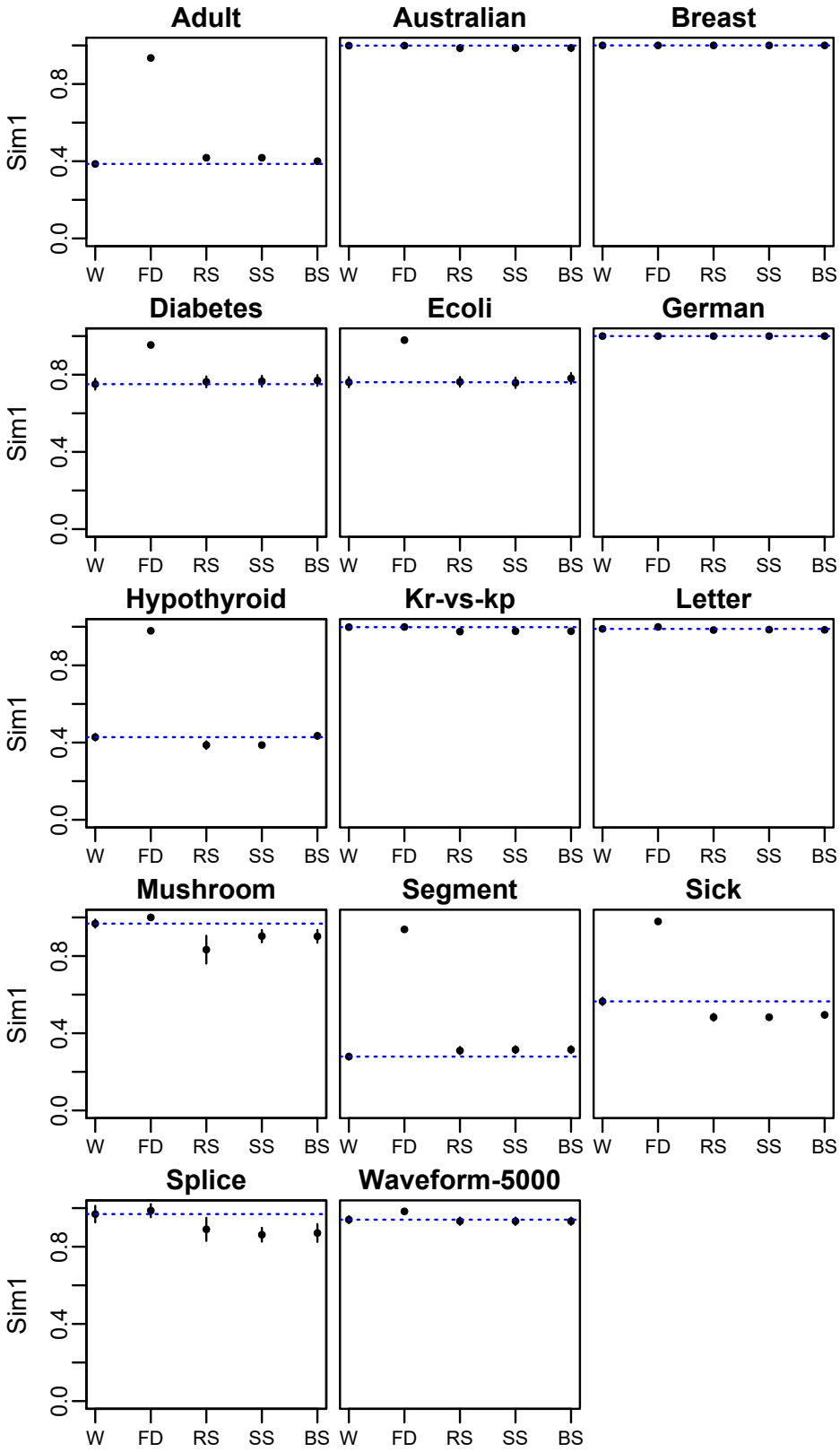


Figure 4.3: Average results of Sim1.
W = Windowing, FD = Full-Dataset, RS = Random-Sampling, SS = Stratified-Sampling, and BS = Balanced-Sampling.
Values closer to one reflect higher similarity in terms of pairs attribute-value.

There are some datasets, like Breast and German, where all the techniques have one as the measured value of $Sim1$. Unfortunately, as in the previous experiment, this notion of similarity neither seems to correlate with the observed accuracy, for instance, as mentioned, for Breast and German all the sampling methods gathers all the original pairs attribute-value ($Sim_1 = 1.0$), but while the accuracy obtained for Breast is around 95%, when using German it is around 71%. In concordance with these results, the window for Breast uses 17% of the available examples, while German uses 64% (Table B.1).

Table B.2 shows the results for the MDL calculated using the test dataset.The construction of the test sets in all the techniques is completely random and stratified. Respecting the number of bits required to encode a tree $L(H)$ (Figure 4.4), Windowing and Full-Dataset tend to induce more complex models, i.e, trees with more nodes. This is probably because Windowing favors the search for more difficult patterns in the set of available instances, which require more complex models to be expressed. Respecting the number of bits required to encode the test data, given the induced decision tree $L(D|H)$ in Figure 4.5, a better compression is achieved using Windowing and Full-Dataset than when using the traditional samplings. Big differences in data compression using Windowing are exhibit in datasets like Mushroom, Segment, and Waveform-5000. One possible explanation for this is that instances gathered by sampling techniques do not capture the data nature because of their random selection and the small number of instances in the sample. The sum of the former metrics, the MDL (4.6), reports bigger models in most of the datasets when using Windowing and Full-Dataset. This result do not represent an advantage, but properties such as the predictive performance also play an important role in model selection.

Table B.3 shows the predictive performance in terms of accuracy and the Area Under the ROC Curve (AUC). Even though, the random, stratified and balanced samplings usually induce simpler models, the decision trees do not seem to be more general than their Windowing and Full-Dataset counterparts. In other words, the predictive ability of decision trees induced with the traditional samplings are, most of the time, lower than the models induced using Windowing and Full-Dataset. Figure 4.7 shows that models induced with Windowing have the same accuracy as those obtained by Full-Dataset and, sometimes, they even show a higher accuracy, e.g., waveform-500. In terms of AUC (4.8), Windowing and Full-Dataset were the best samples,but the balanced sampling is pretty close to their performance.

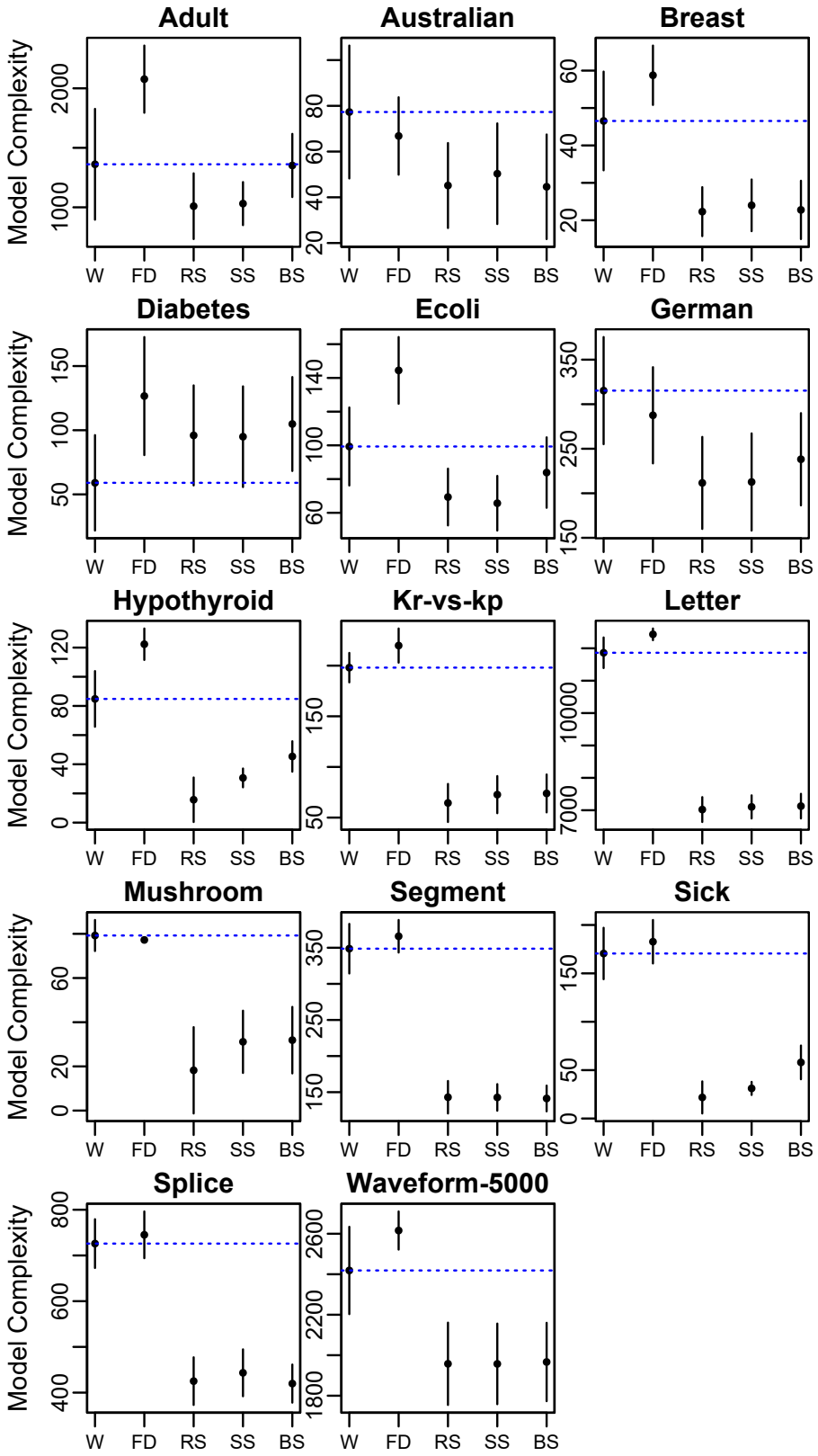


Figure 4.4: Average results of model complexity.
W = Windowing, FD = Full-Dataset, RS = Random-Sampling, SS = Stratified-Sampling, and BS = Balanced-Sampling.
Y-axis is not normalized because model complexity depends on the learning problem. Less complex models are preferred.

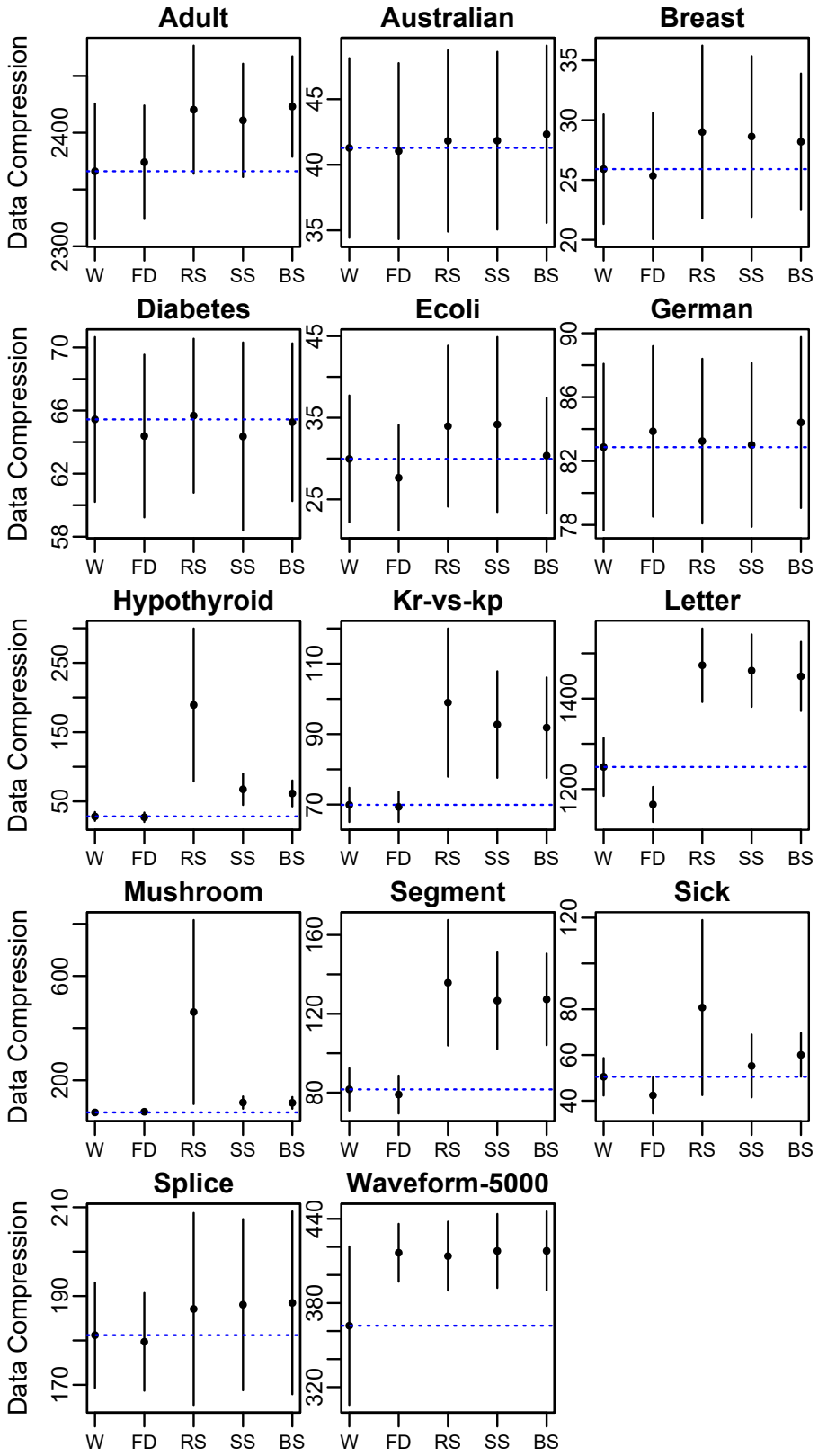


Figure 4.5: Average results of data compression.
W = Windowing, FD = Full-Dataset, RS = Random-Sampling, SS = Stratified-Sampling, and BS = Balanced-Sampling.
Y-axis is not normalized because data compression depends on the learning problem. Higher levels of data compression are preferred.

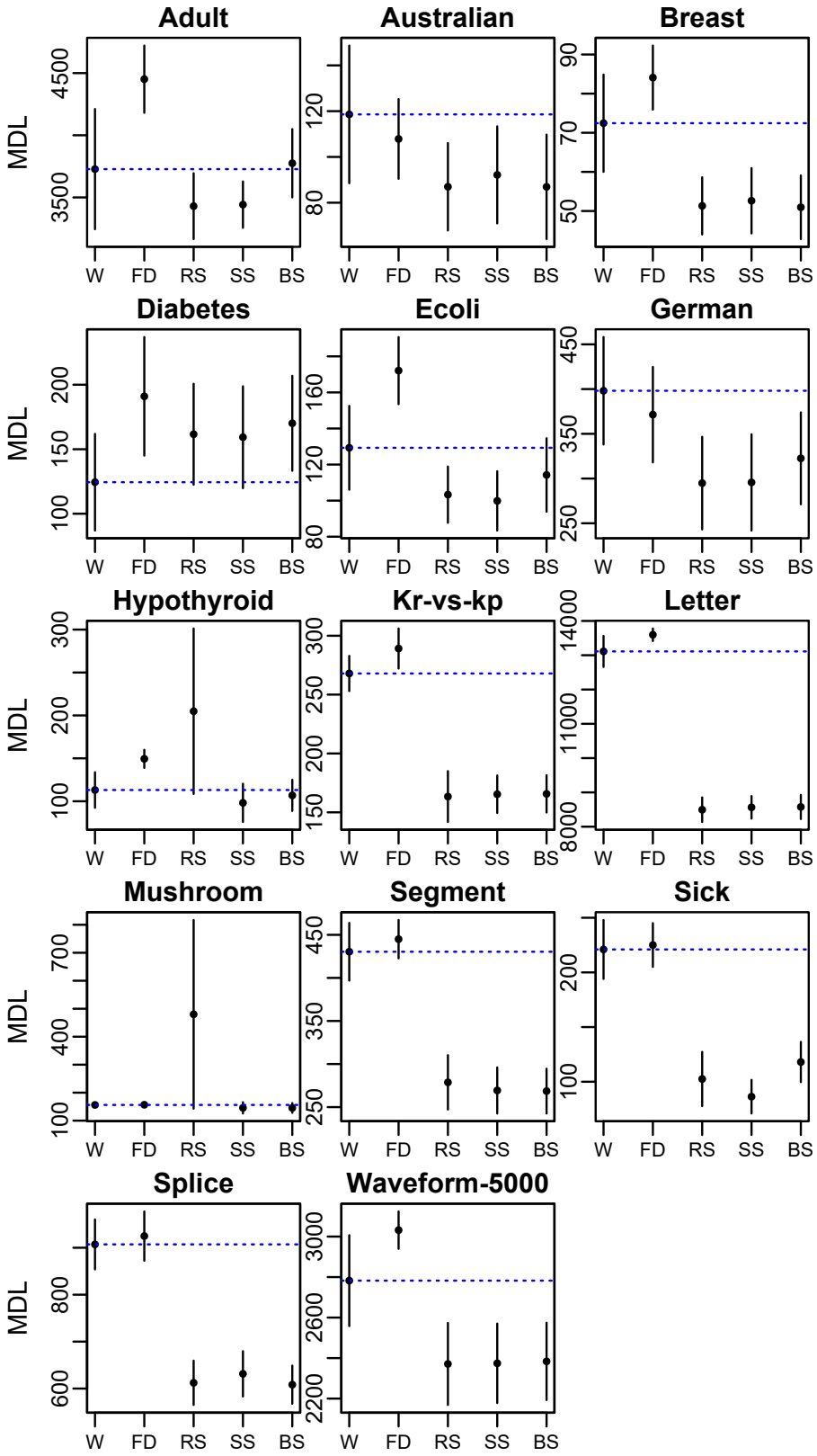


Figure 4.6: Average results of MDL.
W = Windowing, FD = Full-Dataset, RS = Random-Sampling, SS = Stratified-Sampling, and BS = Balanced-Sampling.
Y-axis is not normalized because MDL depends on the learning problem. Lower results of MDL are preferred.

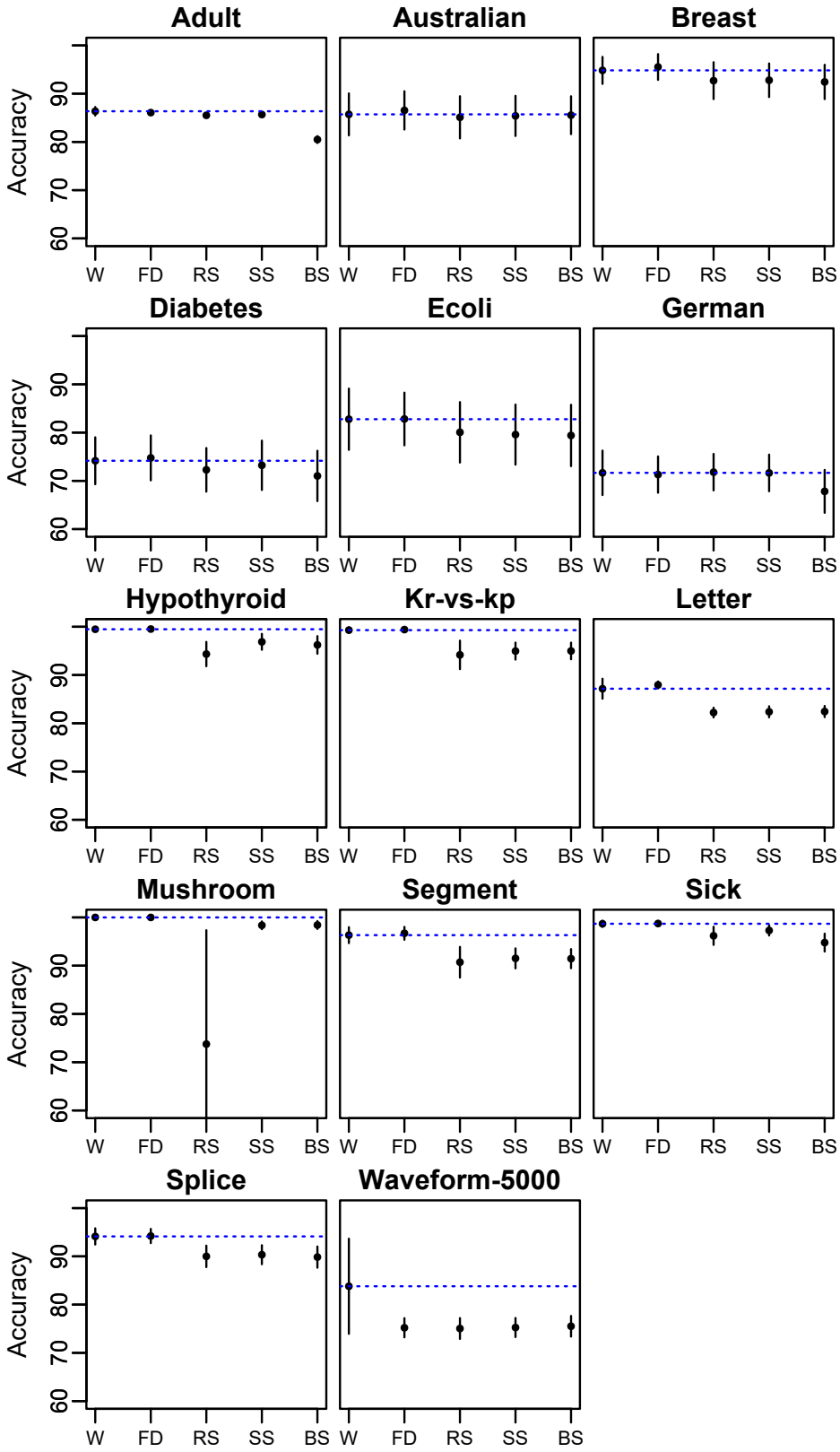


Figure 4.7: Average results of accuracy.
W = Windowing, FD = Full-Dataset, RS = Random-Sampling, SS = Stratified-Sampling, and BS = Balanced-Sampling.
High levels of accuracy are preferred.

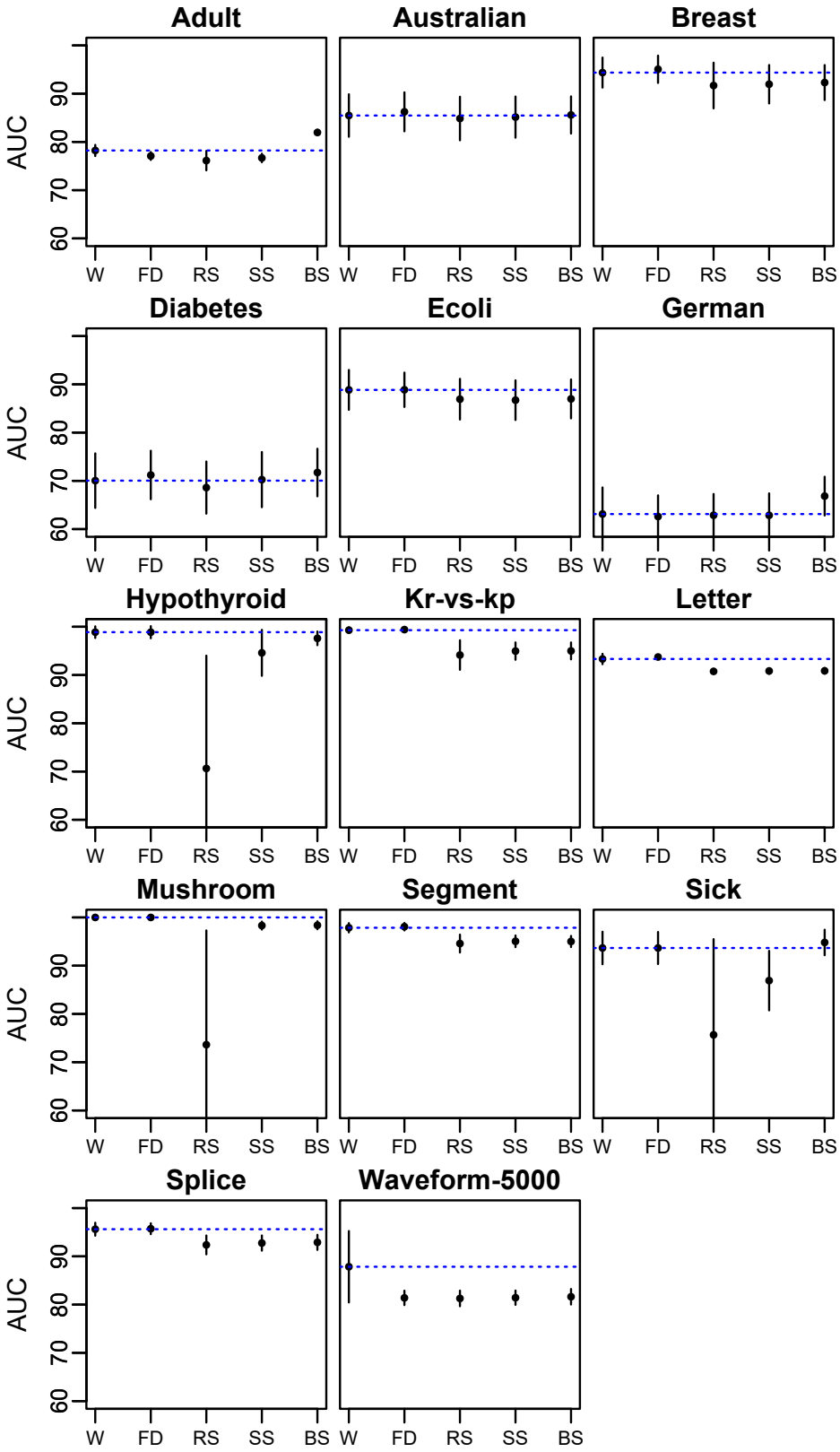


Figure 4.8: Average results of AUC.
W = Windowing, FD = Full-Dataset, RS = Random-Sampling, SS = Stratified-Sampling, and BS = Balanced-Sampling.
High levels of AUC are preferred.

Figure 4.9 shows the results of the post-hoc test in terms of sample properties. In terms of class distribution (Figure 4.9.A), Windowing is known to be the method that tends to skew the distribution the most, given that the counter examples added to the window modify in some way the original distribution. Experiment C studies the window evolution in order to characterize the selection of the examples.

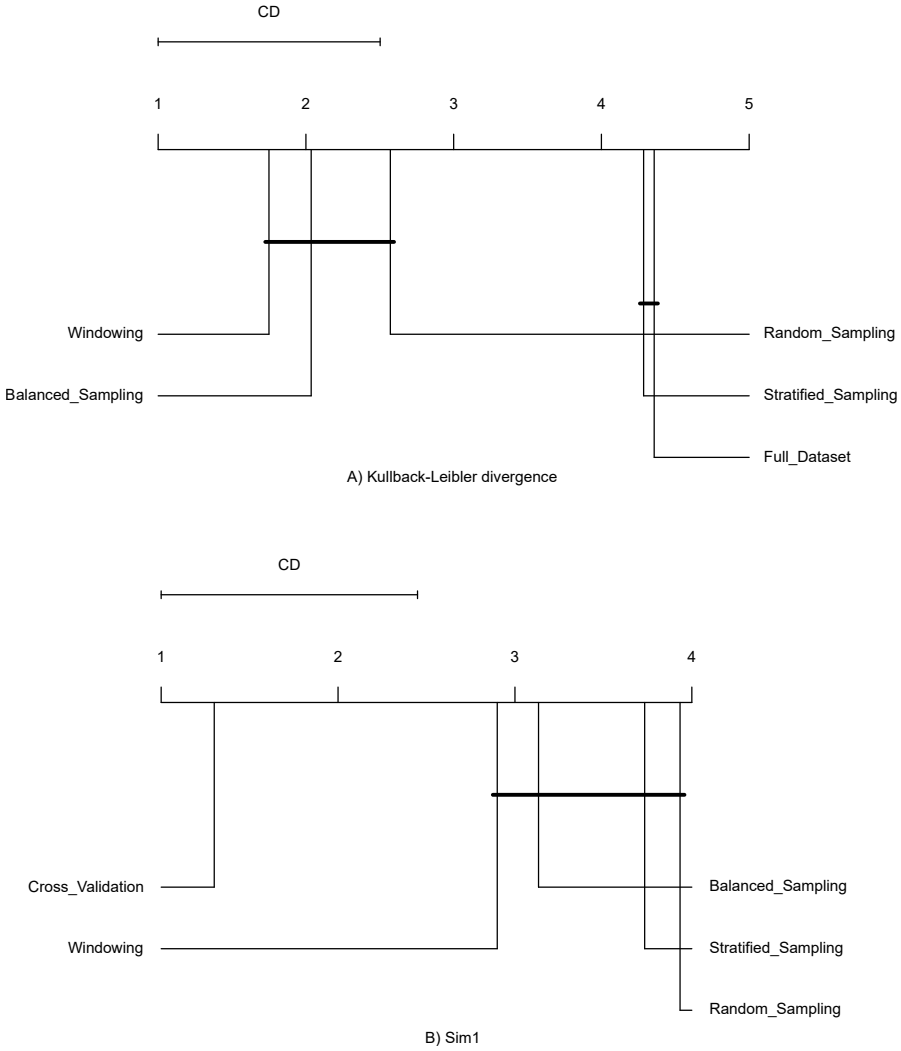


Figure 4.9: Statistical tests for metrics related to dataset features.

As expected, the balanced and the random sampling methods also skew the class distribution showing no significant differences with Windowing. According to the percentage of attribute-value pairs given by Sim_1 (Figure 4.9.B), Windowing and the traditional sampling methods cannot obtain the full set of attribute-value pairs included in the original dataset. Despite this, Windowing is still very competent when it comes to prediction.

Figure 4.10.A shows the results for the number of bits required to encode the induced models ($L(H)$) presented in Table B.2. The groups of connected algorithms are not significantly different. In

this case, the complexity of the models induced using Windowing does not show significant differences with the complexity of the models induced using the Full-Dataset or balanced sampling.

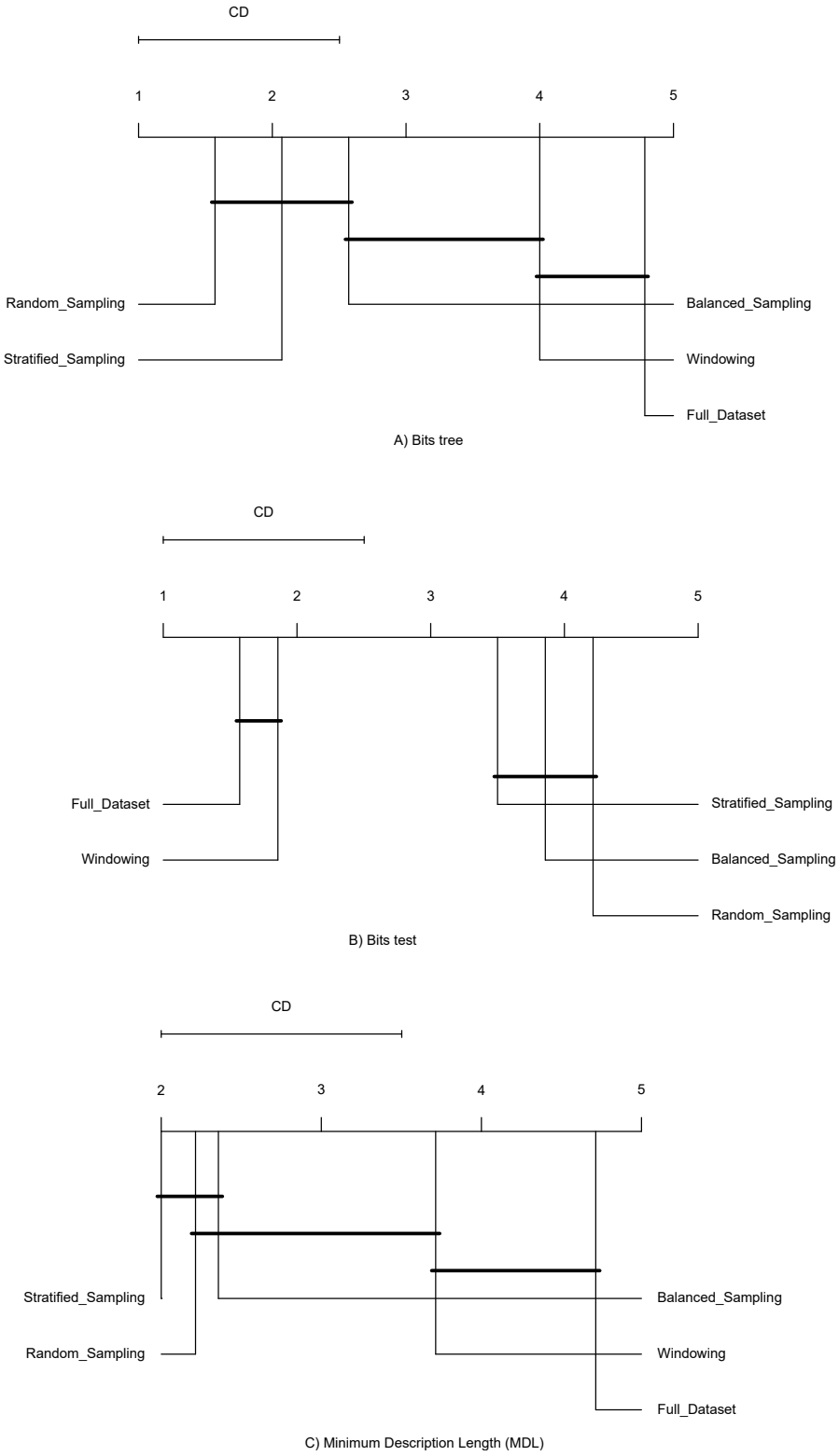


Figure 4.10: Statistical tests for metrics related to MDL.

Figure 4.10.B shows the results in terms of data compression given the decision tree ($L(D|H)$). If the compressibility provided by the models is verified on a stratified sample of unseen data, Windowing and Full-Dataset tend to compress significantly better compared to traditional sampling methods. However, Windowing tends to generate more complex models probably because its heuristic behavior enables the seek for more difficult patterns in the data. Figure 4.10.C shows the results in terms of MDL in the test set. Windowing and Full-Dataset do not show significant differences, nor they are statistically different to the traditional sampling methods. That is, that the induced decision trees generally need the same number of bits to be represented.

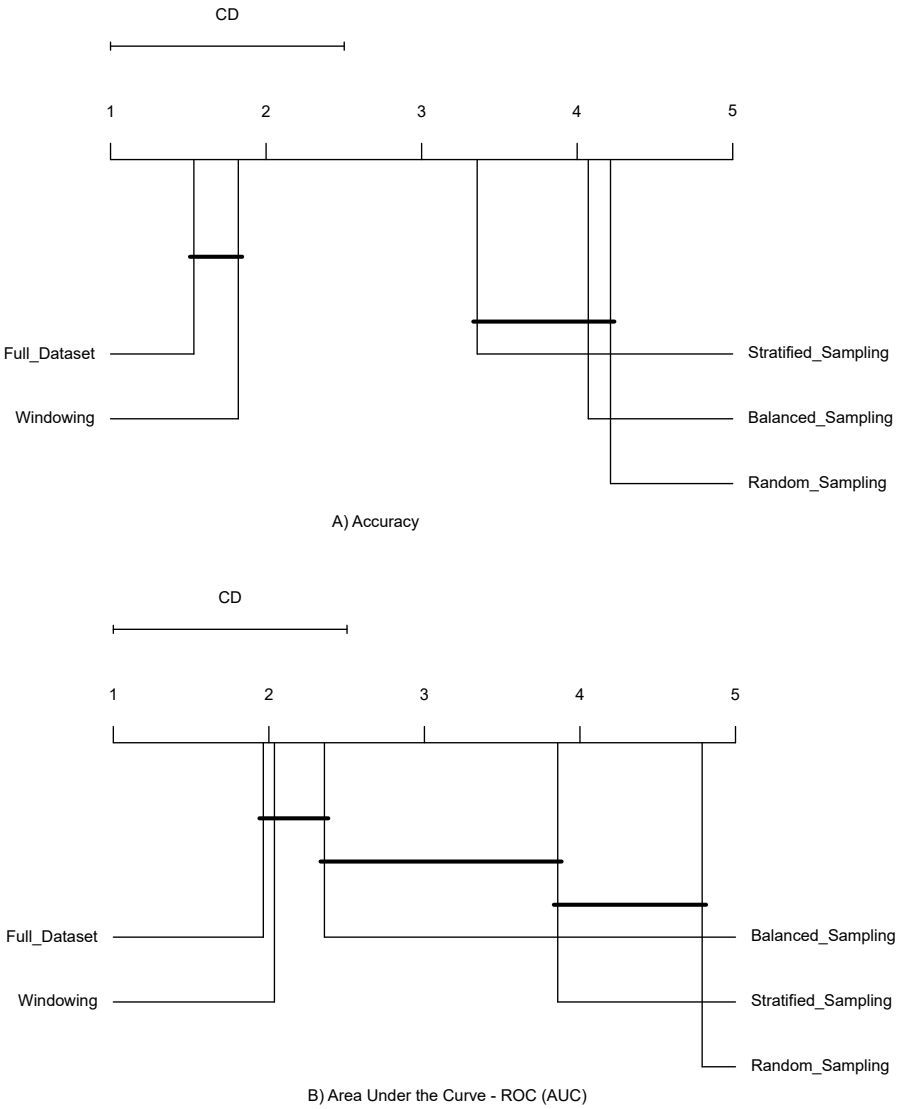


Figure 4.11: Statistical tests for metrics related to predictive performance.

Figure 4.11.A shows the results for accuracy. Windowing performs very well, being almost as accurate as Full-Dataset without significant differences. Both methods are strictly better than the random, balanced, and stratified samplings. When considering the Area

Under the Curve in Figure 4.11.B, results are very similar but the balanced sampling does not show significant differences with Windowing and the Full-Dataset. Recall that both, Windowing and balanced sampling, tend to balance the class distribution.

4.3 Experiment C

Tables on Appendix C reports the information about the cross-validation results per dataset and shows the the evolution of the samplings in terms of the adopted metrics.

The header 'S.D. C.D.' stands for the standard deviation of the class distribution, this suggests a measure of the amount of variation in the classes probability in a dataset. When S.D. C.D. is 0 means that the class distribution is balanced, i.e., there is a equal number of instances per class. The iteration 0 stands for the first sample extracted of the available instances. The figures in this section shows a random-selected Cross-validation fold per algorithm.

The study of the window evolution suggests that Windowing skews the class distribution of some datasets as shown in the results of KL divergence. But these values can not be formally interpreted as a distance since this metric is not a symmetric measure. Figure 4.12 shows that the first windows got coefficients near 0 as a consequence of their stratified nature, however, this did not happened in datasets like Ecoli and Hypothyroid because of their extremely unbalanced class distribution. The results also show that unbalanced datasets have bigger divergence suggesting that Windowing shows a tendency to balance the window.

These observations are corroborated in the changes of S.D. C.D (Figure 4.13), where this metric reports substantial reductions in unbalanced datasets and, it keeps similar in balanced datasets. It would seem that the effectiveness of this behavior depends on the number of available instances for the minority classes and their relevance since class distributions in datasets as Hypothyroid and Sick are less unbalanced but with standard deviations near 0.3.

Regarding the number of instances and the information contents, the results show that the more instances, the higher measures of $Sim1$. This is consistent because the probability of collecting new pair attribute-value increases when there are more instances. On the other hand, the values of Red remain unchanged. This performance supports the idea that a different measure for redundancy should be adopted.

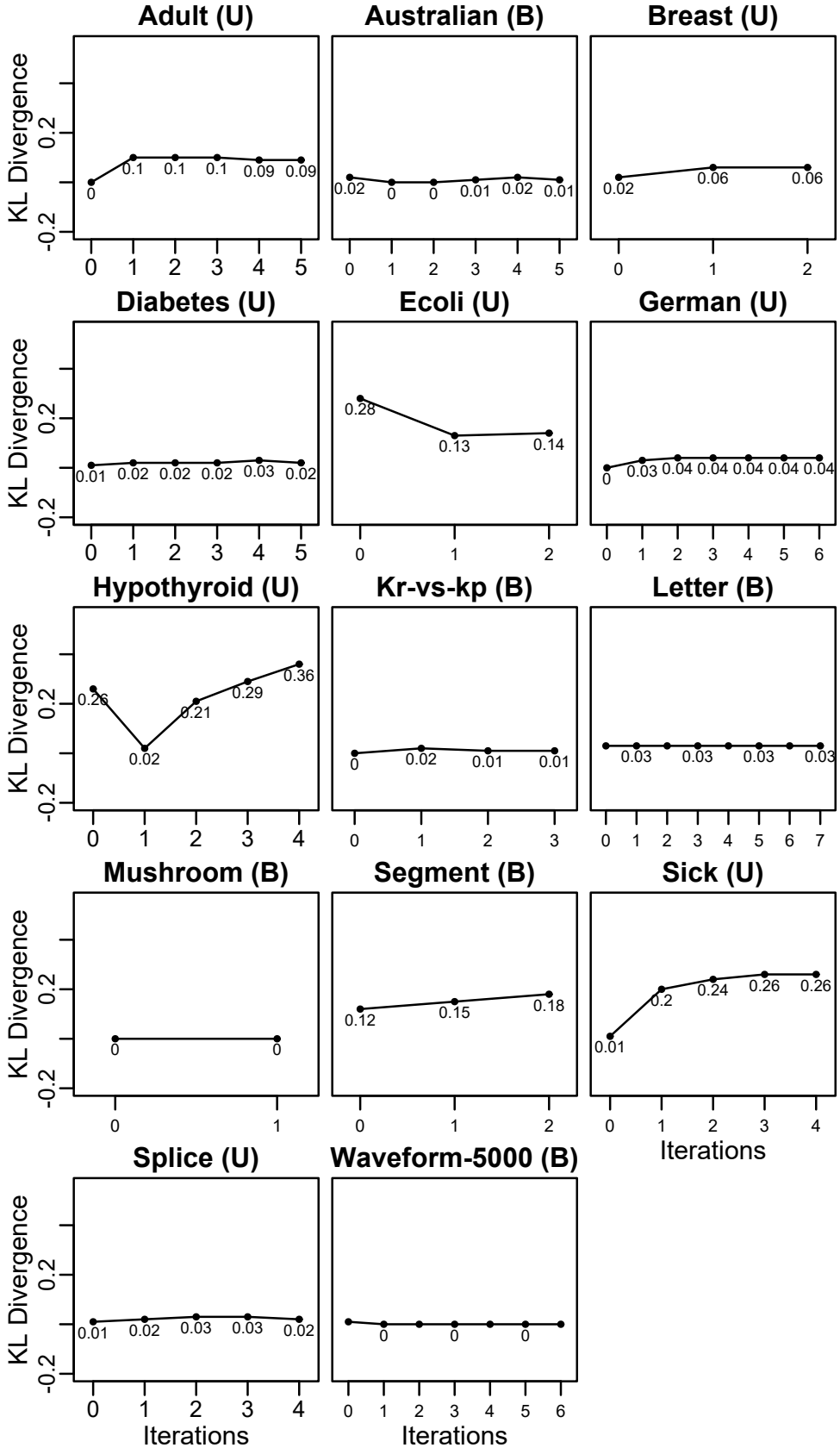


Figure 4.12: Evolution of KL divergence (B = balance dataset, U = unbalanced dataset).

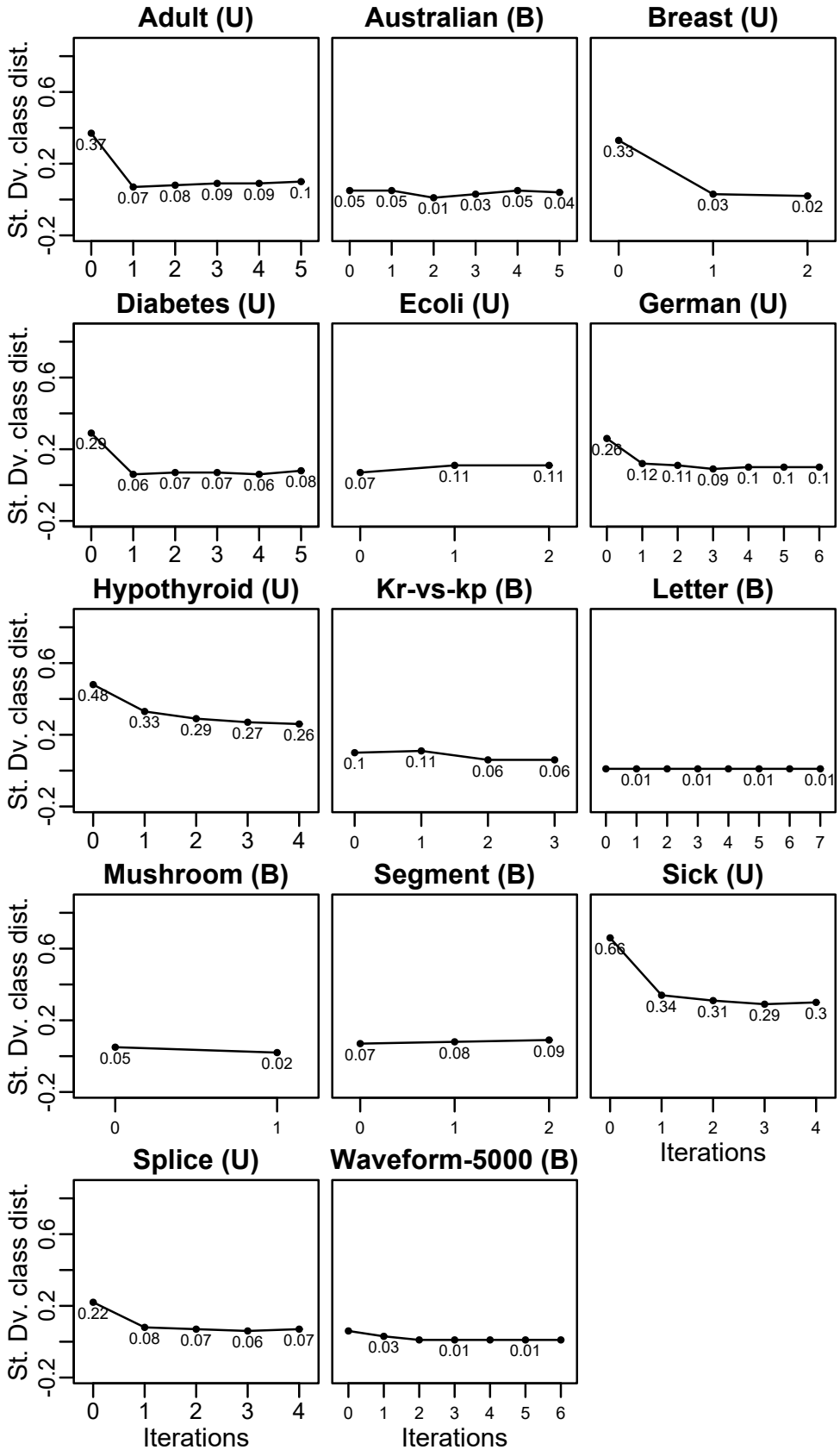


Figure 4.13: Evolution of class distribution (B = balance dataset, U = unbalanced dataset).

The evolution of MDL reports that the models size increments over the iterations, this seems consistent since the decision trees tend to grow when more instances are included in the induction. An interesting behavior was observed in the MDL components, in the first iterations model complexity are small and more bits were required to encode the test dataset (suggesting a poor data compression), but over the iterations model complexity grew, i.e., more nodes were added, and this allowed a better data compression.

In terms of predictive performance, metrics as accuracy and AUC (Figures 4.16 and 4.17), report an increment, but in few occasions the final models was no the most accurate. This encourages to propose new mechanisms to select the most accurate model without losing the search exploration behavior.

Other important observation was that difficult datasets require more iterations to induce better models, and the more iterations the lower predictive performance. Figure 4.14 reports correlation coefficients of -0.56 an -0.50, for Accuracy and AUC, respectively. This suggests a moderated correlation between the number of iterations and the predictive performance. One possible explanations to this, is that the noise in datasets produces more iterations to induce more accurate models, but its performance do not dramatically increment by the noise.

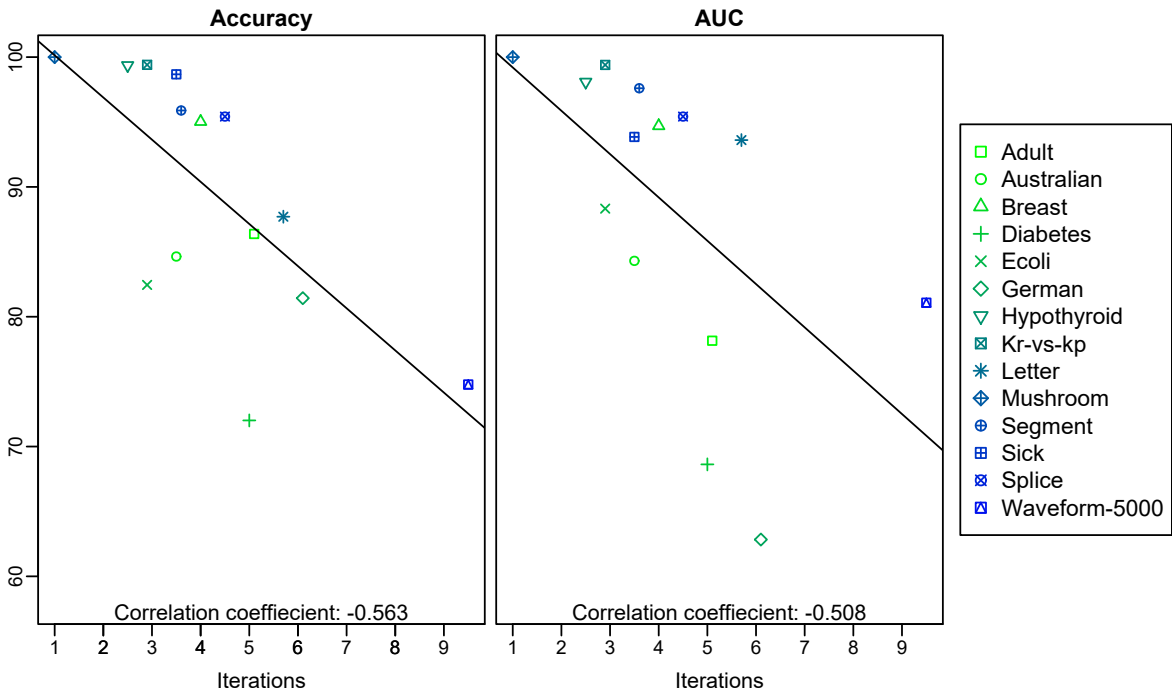


Figure 4.14: Correlation between performance metrics and Windowing iterations.

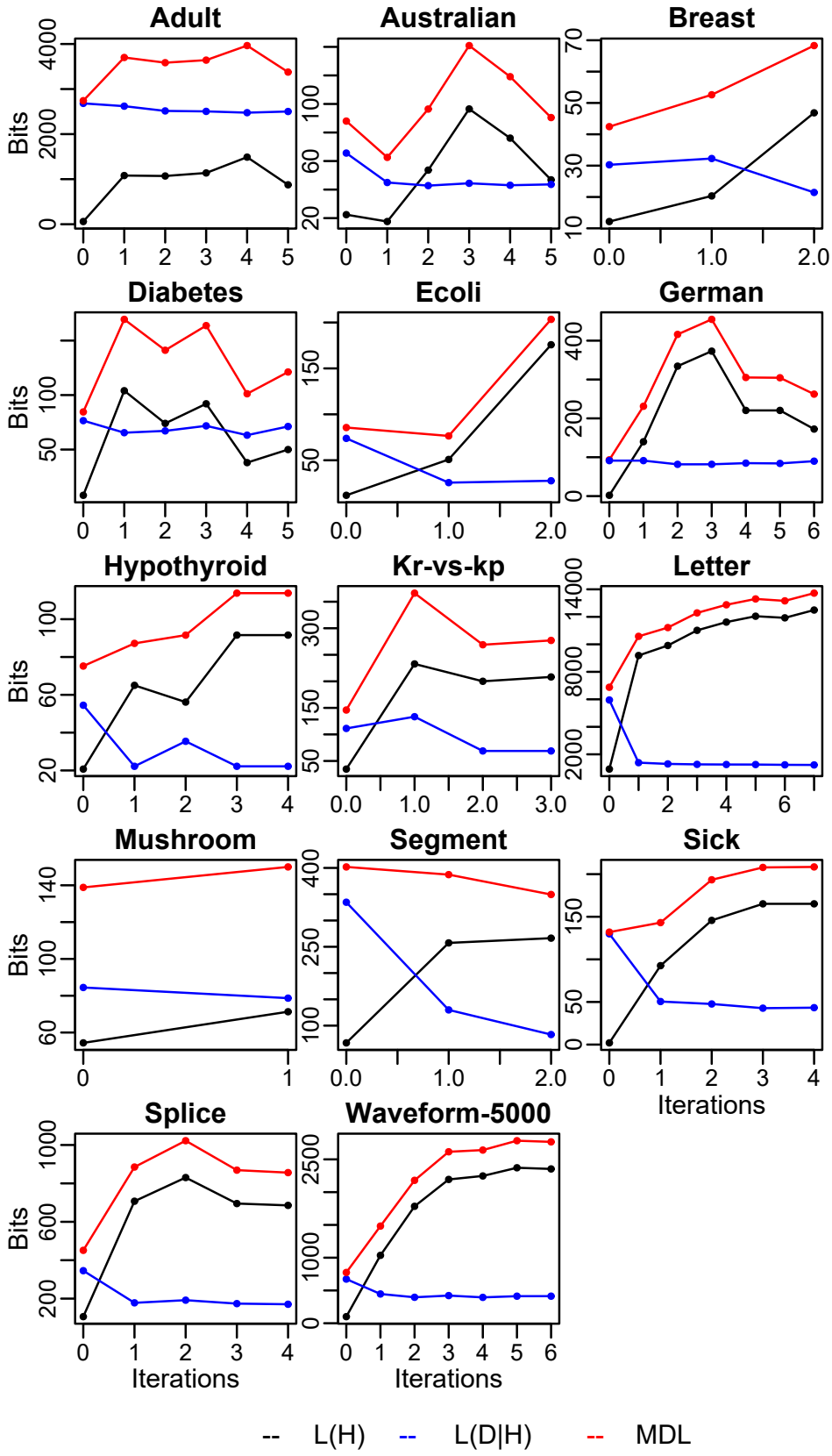


Figure 4.15: Evolution of the MDL metric (B = balance dataset, U = unbalanced dataset).

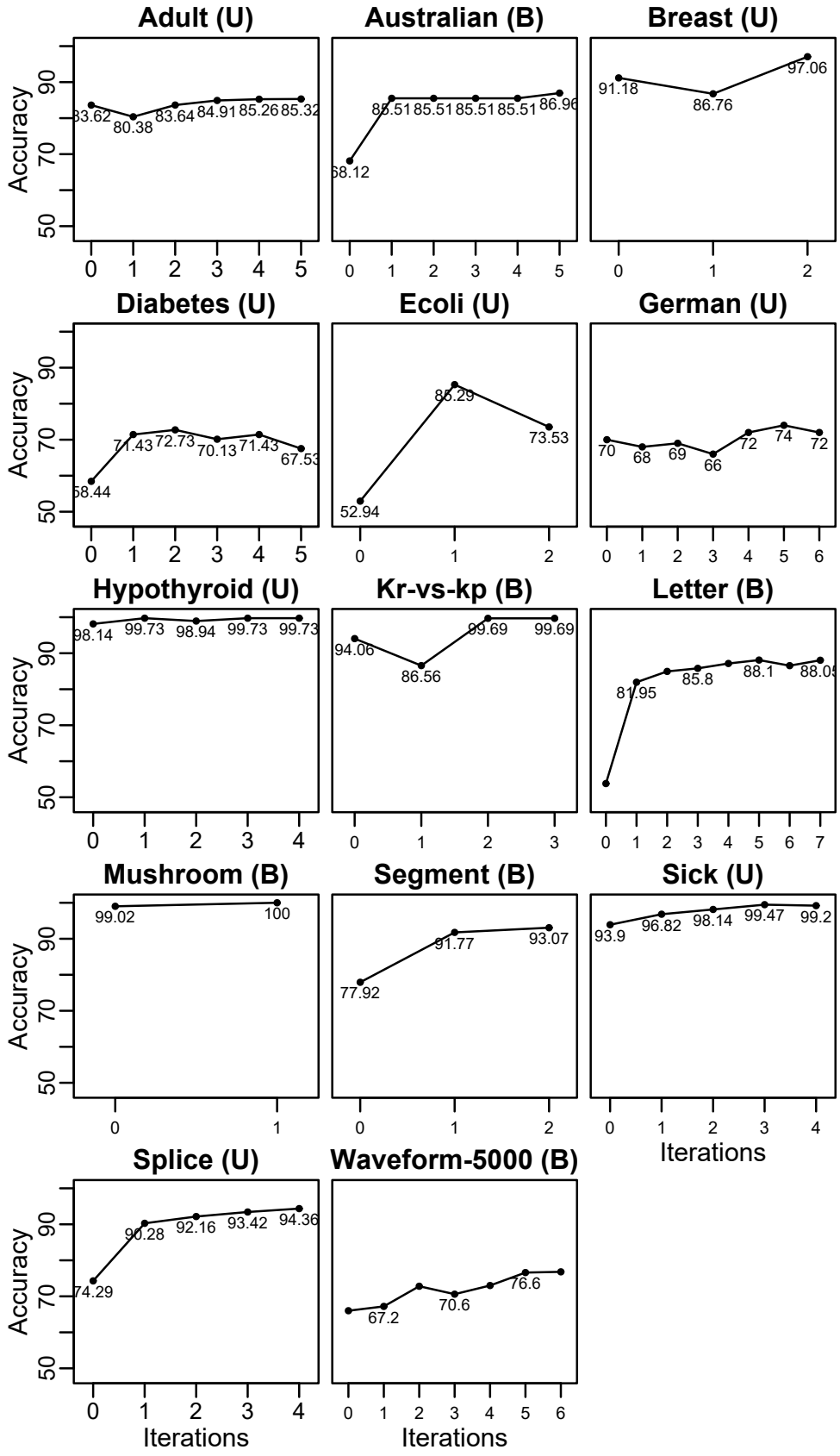


Figure 4.16: Evolution of accuracy (B = balance dataset, U = unbalanced dataset).

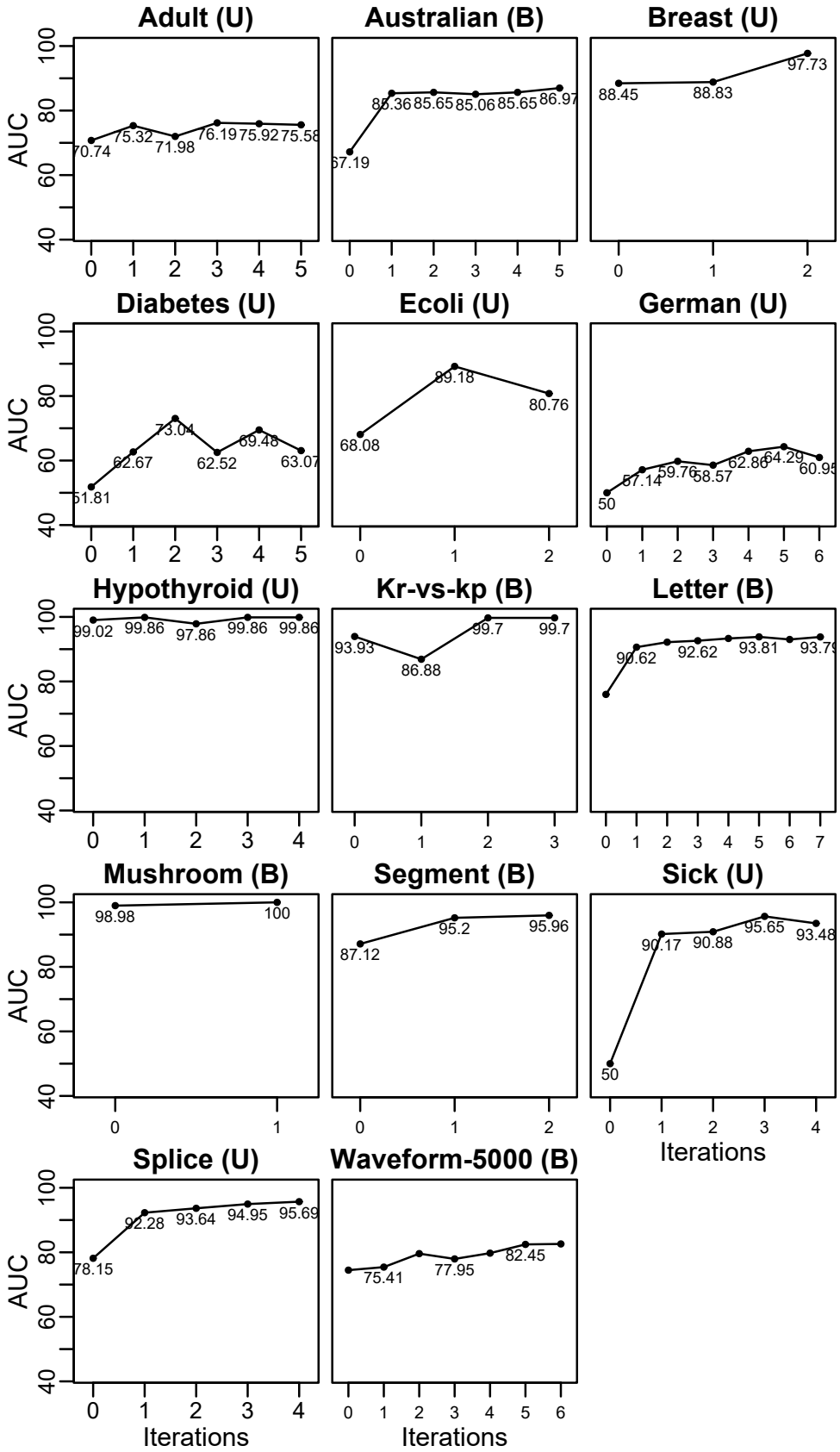


Figure 4.17: Evolution of AUC (B = balance dataset, U = unbalanced dataset).

Conclusions and Future work 5

5.1 Conclusions

Windowing is a method that selects part of the available instances for the induction of divide-and-conquer models like decision trees and association rules, to deal with memory limitations. Although this procedure has not been deeply analyzed due to the fast memory size improvements, this dissertation proposes its use in DMM scenarios under the hypothesis that this technique exhibits consistent behavior using different Machine Learning models. For this, it has been necessary to perform a theoretical study and practical experiments. Regarding the theoretical part, this work surveyed 6 distinct metrics that reflect the features of datasets and models to comprehend the performed sampling.

Windowing not only supplies a natural workflow for collecting distributed data in different scenarios, but it also offers some benefits that supports its use as a sub-sampling method:

- First, the generalization of the behavior of Windowing, beyond decision trees and the J48 algorithm, has been corroborated independently of the inductive method used with Windowing. High levels of accuracy correlate with aggressive samplings up to 5% of the original datasets. Moreover, this behavior suggests that there is not just an adequate classifier to use. The classifiers' and problems' properties have a substantial impact on Windowing performance.
- Second, MDL provided useful information in the sense that, although all methods generate models of similar complexity, it is important to identify which component of the MDL is more relevant in each case. For example, less complex decision trees, as those induced by random, balanced and stratified samplings, are more general but less accurate. In contrast, decision trees with better data compression, such as those induced using Windowing and Full-Dataset, tend to be larger but more accurate. The key factor that makes the difference is the significant reduction of instances for induction. Recall that determining the size of the samples is done automatically in Windowing, based on the autostop condition of this method. To the best of our knowledge, this is the first comparative study of Windowing in this respect.

- Third, even though the Kullback-Leibler divergence and sim_1 do not seem to correlate with accuracy, Windowing shows behavior that favors more balanced class distribution in datasets. This behavior is more visible on unbalanced datasets, but it is restricted by the number of class minority instances and their relevance.
- Fourth, despite the previous works' suggestions for not using Windowing in difficult domains as Diabetes and German. Results report not only a comparable performance but also the setting of an appropriate sample size. The determination of the sample size is an open problem that is tackled most of the time by trial and error.

Windowing can be easily used in distributed environments with tools such as JaCa-DDM, which allows the user to handle large volumes of information and execute Windowing-based techniques. This technique has also shown that is competitive in distributed scenarios using Decision Trees. Although similar results are expected if other classifiers are adopted, experiments must be conducted to verify this. The main difficulty here is adapting some of the metrics, e.g., MDL.

5.2 Future work

This work suggests future lines of research on Windowing, including:

1. Adopting metrics for detecting relevant, noisy, and redundant instances to enhance the quality and size of the obtained samples, in order to improve the performance of the obtained models. Maillo et al. [63] review multiple metrics to describe redundancy, complexity, and density of a problem and also propose two data big metrics. These kind of metrics may be helpful to select instances that provides quality information.
2. Optimizing the search model process. The study of the windows evolution suggests that the last model might not be the most accurate. This problem can be tackled implementing a model memory. Another approach to solve this problem is the use of ensemble techniques, Fürnkranz proposes an ensemble method for rule learning algorithms. This method improves the accuracy of a set of rules trying to induce new rules just from examples that were not correctly classified in the previous iteration.
3. Dealing with datasets of higher number of dimensions. Melgoza-Gutiérrez et al. [64] propose an agent & artifacts-based method to distribute vertical partitions of datasets and deal with the growing time complexity when datasets have

a high number of attributes. It is expected that the achieved understanding on Windowing contributes to combine these approaches.

4. Applying Windowing to real problems. Limón et al. [7] applies Windowing to the segmentation of colposcopic images presenting possible precancerous cervical lesions. Windowing is exploited here to distribute the computational cost of processing a dataset of 1.4×10^6 instances and 30 attributes. The exploitation of Windowing to cope with learning problems of distributed nature is to be explored.

Appendix

Experiment A results A

Table A.1: Accuracies obtained from 10-fold cross validation (na = not available).

Dataset	CV fold	J48	NB	jRip	MM	SMO
adult	1	86.673	84.565	na	na	na
adult	2	85.241	84.033	na	na	na
adult	3	85.545	84.337	na	na	na
adult	4	86.650	85.381	na	na	na
adult	5	86.466	85.319	na	na	na
adult	6	85.954	84.541	na	na	na
adult	7	86.138	83.804	na	na	na
adult	8	86.753	84.848	na	na	na
adult	9	86.712	85.074	na	na	na
adult	10	85.627	83.559	na	na	na
australian	1	86.957	84.058	88.406	82.609	82.609
australian	2	78.261	81.159	79.710	79.710	88.406
australian	3	82.609	85.507	84.058	84.058	85.507
australian	4	84.058	89.855	85.507	86.957	85.507
australian	5	94.203	91.304	91.304	91.304	92.754
australian	6	89.855	89.855	91.304	89.855	88.406
australian	7	79.710	78.261	85.507	72.464	76.812
australian	8	86.957	86.957	84.058	78.261	85.507
australian	9	82.609	82.609	81.159	75.362	81.159
australian	10	86.957	88.406	88.406	76.812	91.304
breast	1	95.652	97.101	97.101	97.101	97.101
breast	2	98.551	100.000	98.551	98.551	100.000
breast	3	94.203	97.101	92.754	98.551	97.101
breast	4	100.000	100.000	95.588	97.059	97.059
breast	5	94.118	97.059	97.059	97.059	97.059
breast	6	89.706	97.059	95.588	94.118	95.588
breast	7	92.647	100.000	98.529	95.588	98.529
breast	8	97.059	95.588	94.118	95.588	95.588
breast	9	95.588	95.588	94.118	92.647	97.059
breast	10	86.765	92.647	89.706	88.235	88.235
diabetes	1	71.429	75.325	79.221	70.130	72.727
diabetes	2	74.026	83.117	74.026	77.922	79.221
diabetes	3	81.818	81.818	84.416	76.623	80.519
diabetes	4	76.623	79.221	72.727	71.429	76.623
diabetes	5	71.429	75.325	66.234	76.623	74.026
diabetes	6	71.429	75.325	74.026	71.429	79.221
diabetes	7	74.026	76.623	55.844	68.831	75.325
diabetes	8	70.130	70.130	67.532	74.026	68.831
diabetes	9	72.368	72.368	72.368	67.105	76.316
diabetes	10	67.105	71.053	71.053	67.105	77.632
ecoli	1	82.353	76.471	79.412	82.353	82.353
ecoli	2	100.000	94.118	91.176	97.059	91.176
ecoli	3	79.412	91.176	94.118	82.353	88.235

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
ecoli	4	82.353	79.412	79.412	82.353	85.294
ecoli	5	76.471	79.412	76.471	79.412	82.353
ecoli	6	79.412	79.412	76.471	76.471	76.471
ecoli	7	87.879	75.758	78.788	72.727	84.848
ecoli	8	81.818	90.909	84.848	87.879	87.879
ecoli	9	78.788	90.909	75.758	87.879	81.818
ecoli	10	78.788	81.818	75.758	72.727	84.848
german	1	80.000	75.000	70.000	76.000	82.000
german	2	70.000	73.000	77.000	74.000	75.000
german	3	64.000	75.000	64.000	69.000	72.000
german	4	70.000	76.000	71.000	73.000	74.000
german	5	68.000	80.000	73.000	68.000	77.000
german	6	79.000	76.000	71.000	64.000	78.000
german	7	74.000	76.000	67.000	73.000	73.000
german	8	71.000	75.000	68.000	72.000	74.000
german	9	64.000	69.000	67.000	66.000	74.000
german	10	71.000	77.000	74.000	61.000	79.000
hypothyroid	1	99.471	95.503	99.471	91.799	93.651
hypothyroid	2	99.206	93.915	99.206	92.857	94.180
hypothyroid	3	99.469	94.960	99.735	95.225	93.369
hypothyroid	4	99.735	94.430	99.469	80.371	94.695
hypothyroid	5	99.469	96.817	99.469	93.899	94.430
hypothyroid	6	99.469	96.021	99.204	93.634	94.960
hypothyroid	7	99.735	96.817	99.469	94.430	93.899
hypothyroid	8	99.469	94.960	99.469	88.064	94.430
hypothyroid	9	99.204	94.430	98.143	93.634	94.430
hypothyroid	10	99.469	95.756	98.674	94.164	94.960
kr-vs-kp	1	99.063	96.875	98.438	98.438	97.188
kr-vs-kp	2	98.438	97.500	96.875	98.438	96.875
kr-vs-kp	3	100.000	95.313	97.500	97.813	96.563
kr-vs-kp	4	100.000	97.500	99.375	99.688	96.250
kr-vs-kp	5	99.375	96.875	99.375	98.750	97.813
kr-vs-kp	6	99.063	97.188	99.375	99.063	96.875
kr-vs-kp	7	99.373	96.865	99.687	98.119	95.298
kr-vs-kp	8	99.060	94.984	98.119	98.746	95.611
kr-vs-kp	9	97.806	96.865	98.119	99.060	97.179
kr-vs-kp	10	99.373	96.552	97.806	99.060	96.552
letter	1	87.050	69.250	85.850	na	na
letter	2	84.950	71.100	85.500	na	na
letter	3	85.150	69.600	86.100	na	na
letter	4	87.450	70.700	83.600	na	na
letter	5	84.100	68.450	84.200	na	na
letter	6	86.350	66.700	83.900	na	na
letter	7	84.400	68.950	86.200	na	na
letter	8	85.750	68.350	85.150	na	na
letter	9	87.550	69.950	86.150	na	na
letter	10	85.250	69.800	86.500	na	na
mushroom	1	100.000	99.508	100.000	100.000	100.000
mushroom	2	100.000	100.000	100.000	100.000	100.000
mushroom	3	100.000	99.754	100.000	100.000	100.000
mushroom	4	100.000	99.877	100.000	100.000	100.000

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
mushroom	5	100.000	99.877	100.000	100.000	100.000
mushroom	6	100.000	99.631	100.000	100.000	100.000
mushroom	7	100.000	99.877	100.000	100.000	100.000
mushroom	8	100.000	100.000	100.000	100.000	100.000
mushroom	9	100.000	99.631	100.000	100.000	100.000
mushroom	10	100.000	99.877	100.000	100.000	100.000
segment	1	94.805	87.013	94.372	95.671	91.775
segment	2	95.238	80.087	95.238	96.104	92.641
segment	3	98.268	83.550	93.939	96.970	94.805
segment	4	96.970	85.714	95.238	93.939	89.610
segment	5	94.805	84.416	93.074	96.537	90.909
segment	6	95.671	83.550	96.970	95.238	90.043
segment	7	98.701	82.684	96.970	97.835	92.641
segment	8	96.104	85.281	95.671	97.403	93.506
segment	9	98.268	85.281	95.671	96.104	95.238
segment	10	96.537	84.848	98.268	95.238	93.074
sick	1	98.148	93.915	97.090	93.915	95.238
sick	2	98.677	95.503	97.619	95.238	97.090
sick	3	98.674	97.082	98.143	95.756	96.817
sick	4	99.204	97.613	98.939	97.082	97.347
sick	5	99.204	96.286	98.939	96.552	97.082
sick	6	98.143	98.408	97.347	96.286	96.817
sick	7	97.613	97.082	97.878	96.817	97.613
sick	8	99.204	96.286	98.408	96.021	97.082
sick	9	98.939	94.164	96.021	95.756	95.491
sick	10	98.674	97.082	98.939	97.082	96.552
splice	1	93.103	94.984	91.536	na	89.655
splice	2	93.417	92.790	90.596	na	91.850
splice	3	94.357	96.238	94.044	na	92.163
splice	4	94.044	95.611	89.342	na	93.417
splice	5	92.476	95.611	94.357	na	92.790
splice	6	94.357	96.552	95.611	na	94.044
splice	7	94.671	96.238	90.596	na	94.357
splice	8	94.671	94.671	94.671	na	91.850
splice	9	94.357	95.298	94.044	na	91.850
splice	10	94.984	95.298	92.790	na	92.163
waveform-5000	1	75.400	82.400	78.400	na	85.800
waveform-5000	2	68.400	79.000	74.000	na	85.600
waveform-5000	3	73.800	81.600	77.600	na	85.400
waveform-5000	4	69.400	81.000	77.800	na	84.400
waveform-5000	5	73.400	84.000	78.600	na	85.400
waveform-5000	6	73.400	83.800	77.400	na	85.200
waveform-5000	7	75.000	84.000	77.400	na	86.000
waveform-5000	8	76.200	84.000	78.400	na	89.400
waveform-5000	9	74.000	82.200	75.000	na	86.000
waveform-5000	10	72.000	81.600	75.600	na	86.200

Table A.2: Percentage of used instances for induction (na = not available).

Dataset	CV fold	J48	NB	jRip	MM	SMO
Adult	1	0.294	0.217	na	na	na
Adult	2	0.300	0.215	na	na	na
Adult	3	0.288	0.217	na	na	na
Adult	4	0.299	0.215	na	na	na
Adult	5	0.300	0.213	na	na	na
Adult	6	0.303	0.213	na	na	na
Adult	7	0.298	0.214	na	na	na
Adult	8	0.301	0.214	na	na	na
Adult	9	0.284	0.212	na	na	na
Adult	10	0.298	0.211	na	na	na
Australian	1	0.333	0.246	0.341	0.412	0.248
Australian	2	0.277	0.232	0.336	0.429	0.261
Australian	3	0.306	0.235	0.294	0.326	0.265
Australian	4	0.329	0.262	0.319	0.435	0.261
Australian	5	0.325	0.249	0.351	0.452	0.288
Australian	6	0.314	0.265	0.377	0.442	0.296
Australian	7	0.283	0.225	0.355	0.371	0.293
Australian	8	0.326	0.252	0.320	0.377	0.290
Australian	9	0.312	0.236	0.296	0.400	0.267
Australian	10	0.330	0.252	0.364	0.345	0.294
Breast	1	0.151	0.053	0.161	0.119	0.105
Breast	2	0.179	0.063	0.146	0.126	0.102
Breast	3	0.161	0.061	0.143	0.117	0.102
Breast	4	0.183	0.061	0.173	0.136	0.100
Breast	5	0.168	0.061	0.124	0.114	0.091
Breast	6	0.164	0.059	0.151	0.100	0.095
Breast	7	0.176	0.061	0.158	0.117	0.097
Breast	8	0.182	0.059	0.177	0.123	0.111
Breast	9	0.148	0.053	0.133	0.120	0.094
Breast	10	0.151	0.051	0.123	0.092	0.072
Diabetes	1	0.570	0.405	0.516	0.441	0.421
Diabetes	2	0.599	0.406	0.572	0.517	0.440
Diabetes	3	0.603	0.424	0.568	0.497	0.452
Diabetes	4	0.568	0.414	0.546	0.482	0.428
Diabetes	5	0.471	0.396	0.493	0.521	0.445
Diabetes	6	0.505	0.398	0.533	0.500	0.452
Diabetes	7	0.548	0.411	0.417	0.421	0.430
Diabetes	8	0.574	0.401	0.513	0.504	0.384
Diabetes	9	0.438	0.400	0.547	0.514	0.413
Diabetes	10	0.551	0.358	0.542	0.449	0.413
Ecoli	1	0.369	0.295	0.393	0.307	0.283
Ecoli	2	0.399	0.289	0.470	0.330	0.295
Ecoli	3	0.378	0.271	0.384	0.271	0.292
Ecoli	4	0.408	0.274	0.420	0.268	0.307
Ecoli	5	0.405	0.250	0.375	0.327	0.330
Ecoli	6	0.336	0.253	0.378	0.307	0.241
Ecoli	7	0.405	0.262	0.446	0.354	0.304
Ecoli	8	0.348	0.274	0.384	0.348	0.321
Ecoli	9	0.333	0.265	0.369	0.339	0.274

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
Ecoli	10	0.408	0.262	0.443	0.292	0.333
German	1	0.573	0.443	0.612	0.602	0.499
German	2	0.574	0.426	0.624	0.587	0.496
German	3	0.561	0.428	0.553	0.522	0.488
German	4	0.589	0.428	0.591	0.591	0.467
German	5	0.599	0.444	0.599	0.591	0.434
German	6	0.582	0.435	0.591	0.597	0.434
German	7	0.567	0.428	0.604	0.589	0.502
German	8	0.472	0.435	0.590	0.573	0.474
German	9	0.588	0.425	0.593	0.575	0.463
German	10	0.480	0.437	0.624	0.575	0.519
Hypothyroid	1	0.045	0.138	0.056	0.270	0.133
Hypothyroid	2	0.049	0.135	0.053	0.231	0.112
Hypothyroid	3	0.043	0.115	0.054	0.247	0.121
Hypothyroid	4	0.050	0.128	0.050	0.248	0.103
Hypothyroid	5	0.046	0.143	0.059	0.224	0.124
Hypothyroid	6	0.046	0.128	0.052	0.220	0.116
Hypothyroid	7	0.044	0.111	0.051	0.253	0.130
Hypothyroid	8	0.049	0.118	0.051	0.214	0.134
Hypothyroid	9	0.043	0.116	0.053	0.268	0.125
Hypothyroid	10	0.052	0.105	0.052	0.241	0.112
Kr-vs-kp	1	0.070	0.159	0.132	0.077	0.120
Kr-vs-kp	2	0.087	0.155	0.113	0.074	0.113
Kr-vs-kp	3	0.079	0.135	0.133	0.085	0.122
Kr-vs-kp	4	0.086	0.155	0.132	0.099	0.115
Kr-vs-kp	5	0.073	0.165	0.124	0.084	0.121
Kr-vs-kp	6	0.083	0.152	0.143	0.090	0.117
Kr-vs-kp	7	0.079	0.162	0.131	0.087	0.123
Kr-vs-kp	8	0.074	0.151	0.130	0.087	0.122
Kr-vs-kp	9	0.078	0.169	0.143	0.096	0.124
Kr-vs-kp	10	0.077	0.160	0.132	0.084	0.127
Letter	1	0.373	0.377	0.407	na	na
Letter	2	0.314	0.386	0.401	na	na
Letter	3	0.345	0.382	0.410	na	na
Letter	4	0.362	0.378	0.384	na	na
Letter	5	0.308	0.384	0.365	na	na
Letter	6	0.383	0.382	0.366	na	na
Letter	7	0.349	0.385	0.402	na	na
Letter	8	0.348	0.382	0.383	na	na
Letter	9	0.387	0.378	0.406	na	na
Letter	10	0.348	0.379	0.409	na	na
Mushroom	1	0.031	0.042	0.032	0.030	0.031
Mushroom	2	0.030	0.040	0.032	0.029	0.030
Mushroom	3	0.031	0.040	0.039	0.029	0.030
Mushroom	4	0.031	0.039	0.038	0.031	0.029
Mushroom	5	0.031	0.040	0.038	0.029	0.029
Mushroom	6	0.032	0.040	0.034	0.030	0.030
Mushroom	7	0.031	0.040	0.034	0.031	0.029
Mushroom	8	0.030	0.042	0.037	0.030	0.030
Mushroom	9	0.031	0.040	0.035	0.030	0.028
Mushroom	10	0.030	0.041	0.032	0.029	0.030

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
Poker-lsn	1	0.046	0.594	na	na	na
Poker-lsn	2	0.046	0.595	na	na	na
Poker-lsn	3	0.045	0.593	na	na	na
Poker-lsn	4	0.047	0.594	na	na	na
Poker-lsn	5	0.044	0.593	na	na	na
Poker-lsn	6	0.046	0.580	na	na	na
Poker-lsn	7	0.047	0.593	na	na	na
Poker-lsn	8	0.049	0.585	na	na	na
Poker-lsn	9	0.050	0.592	na	na	na
Poker-lsn	10	0.046	0.593	na	na	na
Segment	1	0.164	0.217	0.194	0.129	0.194
Segment	2	0.171	0.212	0.206	0.136	0.186
Segment	3	0.164	0.230	0.203	0.155	0.193
Segment	4	0.173	0.230	0.198	0.159	0.183
Segment	5	0.166	0.217	0.209	0.144	0.181
Segment	6	0.168	0.226	0.173	0.159	0.183
Segment	7	0.163	0.215	0.205	0.160	0.192
Segment	8	0.152	0.206	0.180	0.153	0.192
Segment	9	0.157	0.214	0.170	0.153	0.193
Segment	10	0.166	0.228	0.176	0.124	0.184
Sick	1	0.065	0.090	0.089	0.099	0.106
Sick	2	0.078	0.103	0.080	0.113	0.112
Sick	3	0.075	0.117	0.091	0.123	0.117
Sick	4	0.077	0.100	0.092	0.118	0.105
Sick	5	0.079	0.096	0.096	0.131	0.105
Sick	6	0.078	0.092	0.091	0.126	0.111
Sick	7	0.071	0.096	0.084	0.131	0.103
Sick	8	0.074	0.108	0.081	0.124	0.105
Sick	9	0.070	0.104	0.092	0.100	0.103
Sick	10	0.070	0.106	0.089	0.106	0.109
Splice	1	0.264	0.108	0.247	na	0.186
Splice	2	0.261	0.112	0.229	na	0.199
Splice	3	0.243	0.111	0.265	na	0.199
Splice	4	0.261	0.117	0.209	na	0.189
Splice	5	0.260	0.108	0.262	na	0.199
Splice	6	0.253	0.118	0.269	na	0.207
Splice	7	0.266	0.115	0.250	na	0.202
Splice	8	0.267	0.109	0.268	na	0.199
Splice	9	0.248	0.110	0.257	na	0.203
splice	10	0.262	0.112	0.268	na	0.207
Waveform-5000	1	0.601	0.210	0.523	na	0.272
Waveform-5000	2	0.600	0.217	0.537	na	0.263
Waveform-5000	3	0.523	0.207	0.525	na	0.251
Waveform-5000	4	0.607	0.212	0.535	na	0.254
Waveform-5000	5	0.579	0.213	0.525	na	0.267
Waveform-5000	6	0.592	0.226	0.534	na	0.270
Waveform-5000	7	0.609	0.225	0.534	na	0.241
Waveform-5000	8	0.604	0.210	0.530	na	0.272
Waveform-5000	9	0.596	0.206	0.527	na	0.266
Waveform-5000	10	0.610	0.224	0.525	na	0.267

Table A.3: Results of the metric KL divergence (na = not available).

Dataset	CV fold	J48	NB	jRip	MM	SMO
Adult	1	0.094	0.353	na	na	na
Adult	2	0.100	0.372	na	na	na
Adult	3	0.101	0.377	na	na	na
Adult	4	0.097	0.364	na	na	na
Adult	5	0.095	0.367	na	na	na
Adult	6	0.097	0.362	na	na	na
Adult	7	0.100	0.375	na	na	na
Adult	8	0.089	0.383	na	na	na
Adult	9	0.098	0.367	na	na	na
Adult	10	0.099	0.368	na	na	na
Australian	1	0.023	0.006	0.009	0.001	0.007
Australian	2	0.012	0.005	0.004	0.002	0.002
Australian	3	0.012	0.008	0.002	0.006	0.002
Australian	4	0.016	0.004	0.001	0.001	0.009
Australian	5	0.018	0.005	0.014	0.001	0.005
Australian	6	0.021	0.008	0.005	0.000	0.006
Australian	7	0.012	0.014	0.003	0.001	0.002
Australian	8	0.023	0.012	0.006	0.002	0.002
Australian	9	0.026	0.007	0.016	0.004	0.007
Australian	10	0.018	0.007	0.009	0.002	0.006
Breast	1	0.057	0.000	0.037	0.088	0.032
Breast	2	0.042	0.004	0.036	0.055	0.056
Breast	3	0.032	0.013	0.054	0.055	0.056
Breast	4	0.045	0.022	0.077	0.082	0.057
Breast	5	0.062	0.034	0.051	0.074	0.047
Breast	6	0.065	0.021	0.024	0.047	0.052
Breast	7	0.053	0.047	0.032	0.033	0.047
Breast	8	0.075	0.021	0.035	0.055	0.032
Breast	9	0.028	0.001	0.013	0.092	0.069
Breast	10	0.044	0.013	0.017	0.063	0.041
Diabetes	1	0.014	0.037	0.031	0.024	0.047
Diabetes	2	0.009	0.038	0.020	0.029	0.044
Diabetes	3	0.012	0.040	0.016	0.045	0.042
Diabetes	4	0.011	0.046	0.020	0.037	0.047
Diabetes	5	0.020	0.050	0.022	0.027	0.031
Diabetes	6	0.004	0.038	0.013	0.024	0.043
Diabetes	7	0.028	0.042	0.026	0.024	0.036
Diabetes	8	0.014	0.042	0.022	0.022	0.051
Diabetes	9	0.034	0.043	0.034	0.025	0.041
Diabetes	10	0.031	0.042	0.028	0.042	0.039
Ecoli	1	0.105	0.184	0.128	0.157	0.247
Ecoli	2	0.109	0.202	0.094	0.210	0.210
Ecoli	3	0.112	0.152	0.132	0.248	0.263
Ecoli	4	0.117	0.189	0.114	0.138	0.254
Ecoli	5	0.140	0.213	0.123	0.119	0.178
Ecoli	6	0.075	0.171	0.117	0.150	0.256
Ecoli	7	0.116	0.204	0.125	0.181	0.243
Ecoli	8	0.106	0.197	0.118	0.169	0.264
Ecoli	9	0.141	0.195	0.216	0.119	0.270

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
Ecoli	10	0.091	0.186	0.097	0.192	0.196
German	1	0.033	0.062	0.033	0.017	0.037
German	2	0.030	0.061	0.027	0.020	0.039
German	3	0.034	0.060	0.042	0.021	0.034
German	4	0.028	0.057	0.027	0.016	0.039
German	5	0.025	0.052	0.034	0.014	0.041
German	6	0.028	0.052	0.029	0.021	0.045
German	7	0.031	0.065	0.026	0.016	0.035
German	8	0.041	0.058	0.030	0.016	0.038
German	9	0.022	0.054	0.035	0.013	0.041
German	10	0.035	0.063	0.030	0.021	0.027
Hypothyroid	1	0.213	0.260	0.263	0.085	0.483
Hypothyroid	2	0.306	0.284	0.197	0.131	0.582
Hypothyroid	3	0.230	0.371	0.342	0.110	0.575
Hypothyroid	4	0.220	0.313	0.200	0.114	0.601
Hypothyroid	5	0.240	0.250	0.275	0.126	0.553
Hypothyroid	6	0.235	0.342	0.248	0.151	0.541
Hypothyroid	7	0.239	0.327	0.283	0.118	0.483
Hypothyroid	8	0.251	0.307	0.280	0.149	0.428
Hypothyroid	9	0.289	0.407	0.220	0.088	0.456
Hypothyroid	10	0.216	0.373	0.199	0.121	0.573
Kr-vs-kp	1	0.001	0.005	0.036	0.005	0.001
Kr-vs-kp	2	0.002	0.004	0.001	0.000	0.001
Kr-vs-kp	3	0.000	0.003	0.001	0.007	0.001
Kr-vs-kp	4	0.002	0.003	0.000	0.001	0.003
Kr-vs-kp	5	0.001	0.002	0.004	0.000	0.003
Kr-vs-kp	6	0.001	0.003	0.012	0.007	0.002
Kr-vs-kp	7	0.006	0.002	0.005	0.001	0.006
Kr-vs-kp	8	0.009	0.002	0.018	0.008	0.002
Kr-vs-kp	9	0.004	0.001	0.023	0.004	0.004
Kr-vs-kp	10	0.003	0.002	0.010	0.001	0.003
Letter	1	0.027	0.055	0.036	na	na
Letter	2	0.021	0.053	0.033	na	na
Letter	3	0.023	0.053	0.036	na	na
Letter	4	0.025	0.053	0.040	na	na
Letter	5	0.022	0.050	0.034	na	na
Letter	6	0.026	0.055	0.038	na	na
Letter	7	0.027	0.055	0.036	na	na
Letter	8	0.028	0.057	0.034	na	na
Letter	9	0.025	0.054	0.041	na	na
Letter	10	0.026	0.052	0.035	na	na
Mushroom	1	0.013	0.018	0.001	0.002	0.001
Mushroom	2	0.000	0.009	0.000	0.000	0.006
Mushroom	3	0.000	0.017	0.025	0.005	0.013
Mushroom	4	0.001	0.008	0.011	0.000	0.000
Mushroom	5	0.000	0.006	0.005	0.007	0.000
Mushroom	6	0.005	0.003	0.003	0.000	0.002
Mushroom	7	0.000	0.002	0.000	0.002	0.000
Mushroom	8	0.000	0.012	0.001	0.000	0.002
Mushroom	9	0.002	0.006	0.006	0.000	0.001
Mushroom	10	0.006	0.022	0.000	0.000	0.000

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
Poker-lsn	1	0.192	0.009	na	na	na
Poker-lsn	2	0.188	0.010	na	na	na
Poker-lsn	3	0.190	0.010	na	na	na
Poker-lsn	4	0.182	0.010	na	na	na
Poker-lsn	5	0.188	0.010	na	na	na
Poker-lsn	6	0.191	0.008	na	na	na
Poker-lsn	7	0.185	0.010	na	na	na
Poker-lsn	8	0.179	0.009	na	na	na
Poker-lsn	9	0.180	0.009	na	na	na
Poker-lsn	10	0.185	0.009	na	na	na
Segment	1	0.267	0.471	0.182	0.223	0.307
Segment	2	0.265	0.510	0.185	0.267	0.324
Segment	3	0.204	0.516	0.205	0.259	0.333
Segment	4	0.241	0.496	0.305	0.246	0.284
Segment	5	0.235	0.501	0.224	0.257	0.325
Segment	6	0.286	0.441	0.225	0.296	0.302
Segment	7	0.217	0.505	0.229	0.297	0.312
Segment	8	0.223	0.538	0.283	0.290	0.326
Segment	9	0.246	0.476	0.243	0.265	0.359
Segment	10	0.219	0.498	0.205	0.208	0.322
Sick	1	0.270	0.246	0.249	0.241	0.450
Sick	2	0.199	0.171	0.319	0.224	0.485
Sick	3	0.169	0.204	0.263	0.224	0.504
Sick	4	0.258	0.200	0.243	0.245	0.515
Sick	5	0.246	0.205	0.162	0.227	0.497
Sick	6	0.233	0.217	0.273	0.251	0.470
Sick	7	0.162	0.241	0.301	0.225	0.493
Sick	8	0.253	0.180	0.238	0.219	0.495
Sick	9	0.215	0.148	0.287	0.284	0.482
Sick	10	0.197	0.213	0.309	0.265	0.500
Splice	1	0.027	0.010	0.019	na	0.028
Splice	2	0.028	0.024	0.018	na	0.035
Splice	3	0.018	0.028	0.015	na	0.033
Splice	4	0.027	0.019	0.015	na	0.033
Splice	5	0.023	0.008	0.019	na	0.032
Splice	6	0.028	0.018	0.023	na	0.024
Splice	7	0.025	0.017	0.038	na	0.023
Splice	8	0.021	0.008	0.021	na	0.022
Splice	9	0.029	0.020	0.022	na	0.036
Splice	10	0.026	0.015	0.022	na	0.027
Waveform-5000	1	0.000	0.140	0.000	na	0.002
Waveform-5000	2	0.000	0.128	0.000	na	0.002
Waveform-5000	3	0.000	0.160	0.001	na	0.002
Waveform-5000	4	0.000	0.144	0.001	na	0.003
Waveform-5000	5	0.000	0.143	0.001	na	0.002
Waveform-5000	6	0.000	0.144	0.001	na	0.002
Waveform-5000	7	0.000	0.159	0.001	na	0.002
Waveform-5000	8	0.000	0.139	0.001	na	0.002
Waveform-5000	9	0.000	0.163	0.001	na	0.004
Waveform-5000	10	0.000	0.152	0.001	na	0.003

Table A.4: Results of the metric $Sim1$ (na = not available).

Dataset	CV fold	J48	NB	jRip	MM	SMO
Adult	1	0.384	0.291	na	na	na
Adult	2	0.389	0.291	na	na	na
Adult	3	0.375	0.291	na	na	na
Adult	4	0.387	0.292	na	na	na
Adult	5	0.390	0.288	na	na	na
Adult	6	0.393	0.291	na	na	na
Adult	7	0.388	0.290	na	na	na
Adult	8	0.391	0.291	na	na	na
Adult	9	0.374	0.288	na	na	na
Adult	10	0.387	0.288	na	na	na
Australian	1	1.000	1.000	1.000	1.000	1.000
Australian	2	1.000	1.000	1.000	1.000	1.000
Australian	3	1.000	1.000	1.000	1.000	1.000
Australian	4	1.000	1.000	1.000	1.000	1.000
Australian	5	1.000	1.000	1.000	1.000	1.000
Australian	6	1.000	1.000	1.000	1.000	1.000
Australian	7	1.000	1.000	1.000	1.000	1.000
Australian	8	1.000	1.000	1.000	1.000	1.000
Australian	9	1.000	1.000	1.000	1.000	1.000
Australian	10	1.000	1.000	1.000	1.000	1.000
Breast	1	1.000	1.000	1.000	1.000	1.000
Breast	2	1.000	1.000	1.000	1.000	1.000
Breast	3	1.000	1.000	1.000	1.000	1.000
Breast	4	1.000	1.000	1.000	1.000	1.000
Breast	5	1.000	1.000	1.000	1.000	1.000
Breast	6	1.000	1.000	1.000	1.000	1.000
Breast	7	1.000	1.000	1.000	1.000	1.000
Breast	8	1.000	1.000	1.000	1.000	1.000
Breast	9	1.000	1.000	1.000	1.000	1.000
Breast	10	1.000	1.000	1.000	1.000	1.000
Diabetes	1	0.754	0.639	0.720	0.682	0.650
Diabetes	2	0.771	0.622	0.759	0.721	0.644
Diabetes	3	0.774	0.648	0.756	0.711	0.665
Diabetes	4	0.752	0.646	0.746	0.700	0.653
Diabetes	5	0.678	0.618	0.699	0.718	0.655
Diabetes	6	0.693	0.622	0.730	0.715	0.658
Diabetes	7	0.729	0.626	0.634	0.631	0.642
Diabetes	8	0.770	0.627	0.721	0.716	0.605
Diabetes	9	0.650	0.619	0.749	0.714	0.624
Diabetes	10	0.735	0.584	0.721	0.663	0.632
Ecoli	1	0.760	0.660	0.790	0.706	0.652
Ecoli	2	0.784	0.666	0.811	0.695	0.658
Ecoli	3	0.784	0.682	0.776	0.650	0.650
Ecoli	4	0.768	0.650	0.792	0.623	0.660
Ecoli	5	0.798	0.617	0.803	0.757	0.698
Ecoli	6	0.739	0.623	0.776	0.677	0.598
Ecoli	7	0.798	0.644	0.811	0.728	0.677
Ecoli	8	0.730	0.663	0.757	0.728	0.674
Ecoli	9	0.712	0.633	0.722	0.725	0.623

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
Ecoli	10	0.784	0.620	0.814	0.685	0.701
German	1	1.000	1.000	1.000	1.000	1.000
German	2	1.000	1.000	1.000	1.000	1.000
German	3	1.000	1.000	1.000	1.000	1.000
German	4	1.000	1.000	1.000	1.000	1.000
German	5	1.000	1.000	1.000	1.000	0.987
German	6	1.000	1.000	1.000	1.000	1.000
German	7	1.000	1.000	1.000	1.000	1.000
German	8	1.000	1.000	1.000	1.000	1.000
German	9	1.000	1.000	1.000	1.000	1.000
German	10	1.000	1.000	1.000	1.000	1.000
Hypothyroid	1	0.443	0.566	0.496	0.684	0.618
Hypothyroid	2	0.480	0.578	0.488	0.678	0.584
Hypothyroid	3	0.441	0.567	0.492	0.683	0.611
Hypothyroid	4	0.459	0.569	0.472	0.679	0.570
Hypothyroid	5	0.448	0.590	0.515	0.661	0.598
Hypothyroid	6	0.439	0.586	0.489	0.684	0.589
Hypothyroid	7	0.457	0.562	0.484	0.688	0.603
Hypothyroid	8	0.449	0.553	0.494	0.683	0.600
Hypothyroid	9	0.449	0.565	0.474	0.709	0.615
Hypothyroid	10	0.471	0.553	0.486	0.686	0.590
Kr-vs-kp	1	1.000	1.000	1.000	1.000	0.987
Kr-vs-kp	2	1.000	1.000	1.000	1.000	1.000
Kr-vs-kp	3	1.000	1.000	1.000	1.000	1.000
Kr-vs-kp	4	1.000	1.000	1.000	1.000	1.000
Kr-vs-kp	5	0.987	0.987	0.987	0.987	0.987
Kr-vs-kp	6	0.987	1.000	1.000	1.000	1.000
Kr-vs-kp	7	1.000	1.000	1.000	1.000	0.987
Kr-vs-kp	8	1.000	0.987	1.000	1.000	1.000
Kr-vs-kp	9	0.987	1.000	1.000	1.000	1.000
Kr-vs-kp	10	1.000	1.000	1.000	1.000	1.000
Letter	1	0.982	0.982	0.993	na	na
Letter	2	0.986	0.979	0.989	na	na
Letter	3	0.996	0.968	1.000	na	na
Letter	4	0.993	0.972	0.993	na	na
Letter	5	0.979	0.986	0.993	na	na
Letter	6	0.993	0.972	0.989	na	na
Letter	7	0.975	0.972	0.986	na	na
Letter	8	0.982	0.968	0.979	na	na
Letter	9	0.986	0.965	0.979	na	na
Letter	10	0.989	0.979	0.996	na	na
Mushroom	1	0.950	1.000	0.975	0.983	0.966
Mushroom	2	0.983	0.983	0.983	0.992	0.983
Mushroom	3	0.992	0.983	0.992	0.958	0.975
Mushroom	4	0.983	0.992	1.000	0.983	0.992
Mushroom	5	0.950	0.975	0.992	0.975	0.950
Mushroom	6	0.975	1.000	0.975	0.966	0.950
Mushroom	7	0.992	1.000	0.983	0.975	0.975
Mushroom	8	0.966	1.000	0.992	0.983	0.975
Mushroom	9	0.975	0.992	0.975	0.966	0.958
Mushroom	10	0.958	0.992	0.983	0.975	0.983

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
Poker-lsn	1	1.000	1.000	na	na	na
Poker-lsn	2	1.000	1.000	na	na	na
Poker-lsn	3	1.000	1.000	na	na	na
Poker-lsn	4	1.000	1.000	na	na	na
Poker-lsn	5	1.000	1.000	na	na	na
Poker-lsn	6	1.000	1.000	na	na	na
Poker-lsn	7	1.000	1.000	na	na	na
Poker-lsn	8	1.000	1.000	na	na	na
Poker-lsn	9	1.000	1.000	na	na	na
Poker-lsn	10	1.000	1.000	na	na	na
Segment	1	0.277	0.319	0.321	0.237	0.291
Segment	2	0.290	0.316	0.332	0.239	0.282
Segment	3	0.275	0.328	0.325	0.252	0.288
Segment	4	0.291	0.327	0.316	0.276	0.289
Segment	5	0.282	0.319	0.334	0.244	0.280
Segment	6	0.286	0.334	0.290	0.262	0.291
Segment	7	0.279	0.325	0.322	0.262	0.292
Segment	8	0.272	0.311	0.290	0.260	0.297
Segment	9	0.270	0.323	0.286	0.261	0.286
Segment	10	0.279	0.330	0.298	0.227	0.287
Sick	1	0.547	0.565	0.610	0.587	0.599
Sick	2	0.575	0.595	0.567	0.596	0.609
Sick	3	0.569	0.605	0.607	0.610	0.607
Sick	4	0.576	0.568	0.599	0.595	0.575
Sick	5	0.604	0.568	0.613	0.629	0.602
Sick	6	0.597	0.567	0.595	0.637	0.612
Sick	7	0.565	0.583	0.586	0.639	0.575
Sick	8	0.592	0.586	0.585	0.615	0.595
Sick	9	0.552	0.581	0.608	0.577	0.609
Sick	10	0.558	0.599	0.615	0.602	0.616
Splice	1	0.979	0.979	0.983	na	0.979
Splice	2	0.986	0.983	0.990	na	0.983
Splice	3	0.986	0.983	0.990	na	0.983
Splice	4	0.986	0.983	0.983	na	0.986
Splice	5	0.990	0.848	0.986	na	0.983
Splice	6	0.983	0.983	0.986	na	0.983
Splice	7	0.983	0.983	0.986	na	0.986
Splice	8	0.855	0.862	0.866	na	0.841
Splice	9	0.983	0.983	0.990	na	0.983
Splice	10	0.983	0.983	0.979	na	0.979
Waveform-5000	1	0.929	0.707	0.904	na	0.776
Waveform-5000	2	0.928	0.717	0.907	na	0.768
Waveform-5000	3	0.906	0.705	0.906	na	0.756
Waveform-5000	4	0.930	0.710	0.906	na	0.757
Waveform-5000	5	0.923	0.711	0.903	na	0.772
Waveform-5000	6	0.927	0.725	0.907	na	0.773
Waveform-5000	7	0.932	0.728	0.907	na	0.743
Waveform-5000	8	0.929	0.707	0.907	na	0.774
Waveform-5000	9	0.927	0.705	0.905	na	0.769
Waveform-5000	10	0.933	0.725	0.905	na	0.772

Table A.5: Results of the metric *Red* (na = not available).

Dataset	CV fold	J48	NB	jRip	MM	SMO
Adult	1	0.720	0.721	na	na	na
Adult	2	0.720	0.720	na	na	na
Adult	3	0.720	0.721	na	na	na
Adult	4	0.721	0.721	na	na	na
Adult	5	0.720	0.720	na	na	na
Adult	6	0.720	0.720	na	na	na
Adult	7	0.720	0.721	na	na	na
Adult	8	0.721	0.721	na	na	na
Adult	9	0.720	0.721	na	na	na
Adult	10	0.721	0.721	na	na	na
Australian	1	0.615	0.613	0.615	0.617	0.609
Australian	2	0.614	0.606	0.616	0.616	0.612
Australian	3	0.618	0.615	0.623	0.621	0.614
Australian	4	0.606	0.608	0.612	0.609	0.611
Australian	5	0.617	0.612	0.613	0.614	0.614
Australian	6	0.612	0.607	0.612	0.614	0.615
Australian	7	0.609	0.595	0.611	0.611	0.608
Australian	8	0.614	0.609	0.615	0.614	0.612
Australian	9	0.623	0.609	0.609	0.617	0.604
Australian	10	0.618	0.610	0.616	0.611	0.610
Breast	1	0.587	0.577	0.591	0.589	0.571
Breast	2	0.582	0.554	0.581	0.591	0.568
Breast	3	0.593	0.585	0.584	0.584	0.575
Breast	4	0.595	0.576	0.582	0.594	0.581
Breast	5	0.598	0.565	0.586	0.586	0.578
Breast	6	0.602	0.576	0.608	0.593	0.591
Breast	7	0.599	0.565	0.598	0.586	0.567
Breast	8	0.589	0.568	0.599	0.594	0.591
Breast	9	0.610	0.586	0.591	0.592	0.587
Breast	10	0.606	0.607	0.627	0.597	0.605
Diabetes	1	0.580	0.584	0.582	0.583	0.579
Diabetes	2	0.582	0.584	0.581	0.578	0.578
Diabetes	3	0.582	0.581	0.580	0.582	0.582
Diabetes	4	0.581	0.580	0.583	0.581	0.580
Diabetes	5	0.580	0.582	0.581	0.584	0.582
Diabetes	6	0.581	0.582	0.581	0.582	0.582
Diabetes	7	0.582	0.582	0.582	0.584	0.582
Diabetes	8	0.582	0.581	0.582	0.580	0.581
Diabetes	9	0.582	0.581	0.582	0.584	0.582
Diabetes	10	0.581	0.581	0.584	0.582	0.583
Ecoli	1	0.919	0.922	0.919	0.922	0.919
Ecoli	2	0.918	0.922	0.914	0.921	0.919
Ecoli	3	0.918	0.924	0.920	0.922	0.921
Ecoli	4	0.917	0.922	0.918	0.923	0.920
Ecoli	5	0.918	0.907	0.920	0.922	0.920
Ecoli	6	0.917	0.921	0.917	0.919	0.923
Ecoli	7	0.919	0.925	0.918	0.919	0.921
Ecoli	8	0.919	0.924	0.920	0.920	0.921
Ecoli	9	0.921	0.924	0.920	0.920	0.923

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
Ecoli	10	0.918	0.923	0.918	0.922	0.920
German	1	0.618	0.616	0.619	0.618	0.618
German	2	0.618	0.617	0.619	0.619	0.619
German	3	0.621	0.621	0.622	0.619	0.620
German	4	0.617	0.617	0.618	0.620	0.619
German	5	0.616	0.612	0.617	0.618	0.610
German	6	0.618	0.614	0.620	0.620	0.618
German	7	0.619	0.614	0.621	0.617	0.618
German	8	0.621	0.620	0.622	0.620	0.617
German	9	0.617	0.616	0.620	0.618	0.616
German	10	0.617	0.616	0.619	0.616	0.618
Hypothyroid	1	0.844	0.844	0.847	0.843	0.843
Hypothyroid	2	0.845	0.844	0.843	0.841	0.839
Hypothyroid	3	0.839	0.845	0.846	0.842	0.844
Hypothyroid	4	0.844	0.844	0.844	0.843	0.842
Hypothyroid	5	0.842	0.845	0.843	0.841	0.845
Hypothyroid	6	0.847	0.845	0.841	0.844	0.842
Hypothyroid	7	0.844	0.844	0.845	0.840	0.846
Hypothyroid	8	0.845	0.842	0.843	0.841	0.842
Hypothyroid	9	0.847	0.844	0.848	0.843	0.842
Hypothyroid	10	0.838	0.843	0.834	0.841	0.844
Kr-vs-kp	1	0.718	0.721	0.713	0.714	0.716
Kr-vs-kp	2	0.718	0.719	0.711	0.714	0.720
Kr-vs-kp	3	0.717	0.717	0.708	0.710	0.717
Kr-vs-kp	4	0.725	0.719	0.706	0.713	0.719
Kr-vs-kp	5	0.705	0.709	0.702	0.702	0.715
Kr-vs-kp	6	0.701	0.718	0.708	0.712	0.724
Kr-vs-kp	7	0.712	0.717	0.707	0.713	0.709
Kr-vs-kp	8	0.724	0.712	0.709	0.711	0.721
Kr-vs-kp	9	0.717	0.720	0.714	0.711	0.721
Kr-vs-kp	10	0.712	0.719	0.712	0.710	0.722
Letter	1	0.974	0.974	0.974	na	na
Letter	2	0.974	0.974	0.974	na	na
Letter	3	0.974	0.974	0.974	na	na
Letter	4	0.974	0.974	0.974	na	na
Letter	5	0.974	0.974	0.974	na	na
Letter	6	0.974	0.974	0.974	na	na
Letter	7	0.974	0.974	0.974	na	na
Letter	8	0.974	0.974	0.974	na	na
Letter	9	0.974	0.974	0.974	na	na
Letter	10	0.974	0.974	0.974	na	na
Mushroom	1	0.687	0.680	0.697	0.702	0.693
Mushroom	2	0.695	0.684	0.684	0.693	0.689
Mushroom	3	0.693	0.679	0.677	0.705	0.697
Mushroom	4	0.691	0.681	0.697	0.691	0.702
Mushroom	5	0.698	0.679	0.685	0.696	0.693
Mushroom	6	0.691	0.682	0.700	0.696	0.691
Mushroom	7	0.687	0.682	0.694	0.694	0.690
Mushroom	8	0.693	0.684	0.684	0.694	0.701
Mushroom	9	0.690	0.682	0.681	0.690	0.700
Mushroom	10	0.702	0.673	0.685	0.693	0.690

Continued on next page

Dataset	CV fold	J48	NB	jRip	MM	SMO
Poker-lsn	1	0.913	0.914	na	na	na
Poker-lsn	2	0.913	0.914	na	na	na
Poker-lsn	3	0.913	0.914	na	na	na
Poker-lsn	4	0.913	0.914	na	na	na
Poker-lsn	5	0.913	0.914	na	na	na
Poker-lsn	6	0.913	0.914	na	na	na
Poker-lsn	7	0.913	0.914	na	na	na
Poker-lsn	8	0.913	0.914	na	na	na
Poker-lsn	9	0.913	0.914	na	na	na
Poker-lsn	10	0.913	0.914	na	na	na
Segment	1	0.900	0.894	0.900	0.903	0.899
Segment	2	0.899	0.894	0.900	0.903	0.899
Segment	3	0.901	0.894	0.900	0.902	0.898
Segment	4	0.899	0.893	0.898	0.901	0.899
Segment	5	0.901	0.894	0.900	0.903	0.899
Segment	6	0.899	0.895	0.901	0.901	0.899
Segment	7	0.900	0.893	0.900	0.901	0.899
Segment	8	0.900	0.893	0.900	0.902	0.898
Segment	9	0.901	0.893	0.901	0.902	0.898
Segment	10	0.901	0.894	0.901	0.904	0.898
Sick	1	0.660	0.673	0.675	0.670	0.677
Sick	2	0.678	0.678	0.674	0.676	0.682
Sick	3	0.674	0.680	0.672	0.675	0.683
Sick	4	0.677	0.674	0.677	0.672	0.685
Sick	5	0.667	0.674	0.678	0.671	0.676
Sick	6	0.667	0.673	0.669	0.665	0.670
Sick	7	0.669	0.676	0.672	0.674	0.677
Sick	8	0.678	0.681	0.666	0.671	0.677
Sick	9	0.678	0.681	0.677	0.676	0.678
Sick	10	0.671	0.680	0.671	0.676	0.673
Splice	1	0.714	0.709	0.713	na	0.712
Splice	2	0.715	0.711	0.715	na	0.713
Splice	3	0.714	0.711	0.714	na	0.713
Splice	4	0.715	0.711	0.714	na	0.713
Splice	5	0.715	0.682	0.713	na	0.713
Splice	6	0.715	0.710	0.714	na	0.713
Splice	7	0.714	0.711	0.714	na	0.713
Splice	8	0.687	0.685	0.688	na	0.682
Splice	9	0.714	0.711	0.715	na	0.713
Splice	10	0.714	0.710	0.713	na	0.712
Waveform-5000	1	0.694	0.702	0.695	na	0.702
Waveform-5000	2	0.694	0.702	0.694	na	0.702
Waveform-5000	3	0.695	0.702	0.695	na	0.703
Waveform-5000	4	0.694	0.702	0.695	na	0.702
Waveform-5000	5	0.694	0.702	0.695	na	0.702
Waveform-5000	6	0.694	0.701	0.695	na	0.702
Waveform-5000	7	0.694	0.701	0.695	na	0.703
Waveform-5000	8	0.694	0.702	0.695	na	0.702
Waveform-5000	9	0.694	0.702	0.695	na	0.702
Waveform-5000	10	0.694	0.701	0.695	na	0.702

Experiment B results

B

Table B.1: Sample properties.

Dataset	Method	Instances		KL Div		Sim1	
Adult	Windowing	14502.840 ±	574.266	0.128 ±	0.004	0.386 ±	0.012
Adult	Full-Dataset	43957.800 ±	0.402	0.000 ±	0.000	0.935 ±	0.001
Adult	Random-sampling	14502.840 ±	574.266	0.005 ±	0.005	0.418 ±	0.013
Adult	Stratified-sampling	14502.840 ±	574.266	0.000 ±	0.000	0.418 ±	0.013
Adult	Balanced-sampling	14502.840 ±	574.266	0.206 ±	0.000	0.400 ±	0.013
Australian	Windowing	215.440 ±	14.363	0.017 ±	0.008	0.999 ±	0.006
Australian	Full-Dataset	621.000 ±	0.000	0.000 ±	0.000	0.999 ±	0.005
Australian	Random-sampling	215.440 ±	14.363	0.004 ±	0.005	0.986 ±	0.016
Australian	Stratified-sampling	215.440 ±	14.363	0.000 ±	0.000	0.986 ±	0.016
Australian	Balanced-sampling	215.440 ±	14.363	0.009 ±	0.000	0.987 ±	0.016
Breast	Windowing	109.210 ±	14.732	0.086 ±	0.031	1.000 ±	0.000
Breast	Full-Dataset	614.700 ±	0.461	0.000 ±	0.000	1.000 ±	0.000
Breast	Random-sampling	109.210 ±	14.732	0.019 ±	0.017	1.000 ±	0.000
Breast	Stratified-sampling	109.210 ±	14.732	0.000 ±	0.000	1.000 ±	0.000
Breast	Balanced-sampling	109.210 ±	14.732	0.066 ±	0.003	1.000 ±	0.000
Diabetes	Windowing	436.260 ±	27.768	0.025 ±	0.009	0.751 ±	0.028
Diabetes	Full-Dataset	691.200 ±	0.402	0.000 ±	0.000	0.954 ±	0.004
Diabetes	Random-sampling	436.260 ±	27.768	0.001 ±	0.001	0.763 ±	0.028
Diabetes	Stratified-sampling	436.260 ±	27.768	0.000 ±	0.000	0.766 ±	0.028
Diabetes	Balanced-sampling	436.260 ±	27.768	0.067 ±	0.001	0.770 ±	0.028
Ecoli	Windowing	126.640 ±	8.579	0.182 ±	0.055	0.761 ±	0.026
Ecoli	Full-Dataset	302.400 ±	0.492	0.001 ±	0.001	0.979 ±	0.006
Ecoli	Random-sampling	126.640 ±	8.579	0.007 ±	0.010	0.763 ±	0.025
Ecoli	Stratified-sampling	126.640 ±	8.579	0.013 ±	0.003	0.758 ±	0.027
Ecoli	Balanced-sampling	126.640 ±	8.579	0.113 ±	0.028	0.781 ±	0.028
German	Windowing	584.750 ±	25.308	0.041 ±	0.006	1.000 ±	0.000
German	Full-Dataset	900.000 ±	0.000	0.000 ±	0.000	1.000 ±	0.000
German	Random-sampling	584.750 ±	25.308	0.001 ±	0.001	1.000 ±	0.000
German	Stratified-sampling	584.750 ±	25.308	0.000 ±	0.000	1.000 ±	0.000
German	Balanced-sampling	584.750 ±	25.308	0.079 ±	0.015	1.000 ±	0.000
Hypothyroid	Windowing	151.680 ±	9.619	0.262 ±	0.047	0.428 ±	0.017
Hypothyroid	Full-Dataset	3394.800 ±	0.402	0.000 ±	0.000	0.979 ±	0.005
Hypothyroid	Random-sampling	151.680 ±	9.619	0.212 ±	0.103	0.387 ±	0.020
Hypothyroid	Stratified-sampling	151.680 ±	9.619	0.000 ±	0.001	0.387 ±	0.013
Hypothyroid	Balanced-sampling	151.680 ±	9.619	0.668 ±	0.023	0.435 ±	0.016
Kr-vs-kp	Windowing	242.550 ±	18.425	0.010 ±	0.012	0.998 ±	0.004
Kr-vs-kp	Full-Dataset	2876.400 ±	0.492	0.000 ±	0.000	0.999 ±	0.004
Kr-vs-kp	Random-sampling	242.550 ±	18.425	0.106 ±	0.099	0.975 ±	0.013
Kr-vs-kp	Stratified-sampling	242.550 ±	18.425	0.000 ±	0.000	0.977 ±	0.009
Kr-vs-kp	Balanced-sampling	242.550 ±	18.425	0.001 ±	0.000	0.977 ±	0.008
Letter	Windowing	7390.450 ±	491.435	0.037 ±	0.002	0.989 ±	0.006
Letter	Full-Dataset	18000.000 ±	0.000	0.000 ±	0.000	0.999 ±	0.002
Letter	Random-sampling	7390.450 ±	491.435	0.022 ±	0.009	0.983 ±	0.008

Continued on next page

Dataset	Method	Instances		KL Div		Sim1	
Letter	Stratified-sampling	7390.450 ±	491.435	0.000 ±	0.000	0.985 ±	0.007
Letter	Balanced-sampling	7390.450 ±	491.435	0.001 ±	0.000	0.984 ±	0.006
Mushroom	Windowing	219.490 ±	16.871	0.004 ±	0.005	0.968 ±	0.021
Mushroom	Full-Dataset	7311.600 ±	0.492	0.000 ±	0.000	1.000 ±	0.000
Mushroom	Random-sampling	219.490 ±	16.871	2.083 ±	1.852	0.833 ±	0.072
Mushroom	Stratified-sampling	219.490 ±	16.871	0.000 ±	0.000	0.903 ±	0.032
Mushroom	Balanced-sampling	219.490 ±	16.871	0.001 ±	0.000	0.902 ±	0.033
Segment	Windowing	371.280 ±	27.458	0.390 ±	0.076	0.279 ±	0.015
Segment	Full-Dataset	2079.000 ±	0.000	0.000 ±	0.000	0.938 ±	0.003
Segment	Random-sampling	371.280 ±	27.458	0.105 ±	0.144	0.310 ±	0.019
Segment	Stratified-sampling	371.280 ±	27.458	0.000 ±	0.000	0.315 ±	0.018
Segment	Balanced-sampling	371.280 ±	27.458	0.000 ±	0.000	0.315 ±	0.018
Sick	Windowing	264.600 ±	17.420	0.233 ±	0.032	0.565 ±	0.019
Sick	Full-Dataset	3394.800 ±	0.402	0.000 ±	0.000	0.979 ±	0.005
Sick	Random-sampling	264.600 ±	17.420	0.102 ±	0.124	0.483 ±	0.018
Sick	Stratified-sampling	264.600 ±	17.420	0.000 ±	0.000	0.483 ±	0.014
Sick	Balanced-sampling	264.600 ±	17.420	0.665 ±	0.002	0.495 ±	0.014
Splice	Windowing	835.300 ±	29.689	0.036 ±	0.009	0.969 ±	0.043
Splice	Full-Dataset	2871.000 ±	0.000	0.000 ±	0.000	0.987 ±	0.034
Splice	Random-sampling	835.300 ±	29.689	0.014 ±	0.013	0.890 ±	0.060
Splice	Stratified-sampling	835.300 ±	29.689	0.000 ±	0.000	0.862 ±	0.036
Splice	Balanced-sampling	835.300 ±	29.689	0.104 ±	0.001	0.871 ±	0.046
Waveform-5000	Windowing	3263.590 ±	330.000	0.000 ±	0.000	0.940 ±	0.018
Waveform-5000	Full-Dataset	4500.000 ±	0.000	0.000 ±	0.000	0.983 ±	0.001
Waveform-5000	Random-sampling	3263.590 ±	330.000	0.002 ±	0.002	0.932 ±	0.019
Waveform-5000	Stratified-sampling	3263.590 ±	330.000	0.000 ±	0.000	0.932 ±	0.019
Waveform-5000	Balanced-sampling	3263.590 ±	330.000	0.000 ±	0.000	0.932 ±	0.019

Table B.2: Model complexity and test data compression.

Dataset	Method	L(H)		L(D | H)		MDL	
Adult	Windowing	1361.599 ±	465.850	2366.019 ±	59.709	3727.618 ±	483.653
Adult	Cross-Validation	2077.010 ±	282.565	2374.002 ±	49.985	4451.012 ±	270.561
Adult	Random-sampling	1009.386 ±	276.429	2420.278 ±	56.458	3429.664 ±	264.703
Adult	Stratified-sampling	1031.172 ±	181.155	2410.870 ±	49.932	3442.042 ±	186.437
Adult	Balanced-sampling	1351.736 ±	265.668	2423.024 ±	44.271	3774.759 ±	274.906
Australian	Windowing	77.299 ±	29.067	41.284 ±	6.849	118.582 ±	30.088
Australian	Cross-Validation	66.820 ±	16.934	41.044 ±	6.711	107.864 ±	17.430
Australian	Random-sampling	45.151 ±	18.592	41.820 ±	6.916	86.971 ±	19.120
Australian	Stratified-sampling	50.313 ±	22.016	41.836 ±	6.776	92.149 ±	21.220
Australian	Balanced-sampling	44.603 ±	22.878	42.327 ±	6.764	86.929 ±	22.830
Breast	Windowing	46.541 ±	13.199	25.904 ±	4.584	72.445 ±	12.435
Breast	Cross-Validation	58.757 ±	7.942	25.338 ±	5.280	84.095 ±	8.195
Breast	Random-sampling	22.301 ±	6.555	29.008 ±	7.229	51.309 ±	7.316
Breast	Stratified-sampling	23.991 ±	6.915	28.631 ±	6.720	52.622 ±	8.350
Breast	Balanced-sampling	22.767 ±	7.801	28.191 ±	5.710	50.959 ±	8.137
Diabetes	Windowing	59.000 ±	37.207	65.437 ±	5.227	124.437 ±	37.477
Diabetes	Cross-Validation	126.620 ±	46.019	64.383 ±	5.161	191.003 ±	45.988
Diabetes	Random-sampling	95.960 ±	38.989	65.674 ±	4.884	161.634 ±	39.119
Diabetes	Stratified-sampling	94.940 ±	39.261	64.354 ±	5.965	159.294 ±	39.505
Diabetes	Balanced-sampling	104.840 ±	36.621	65.263 ±	5.003	170.103 ±	36.829
Ecoli	Windowing	99.328 ±	23.152	29.959 ±	7.767	129.287 ±	23.257
Ecoli	Cross-Validation	144.454 ±	19.804	27.648 ±	6.460	172.102 ±	18.623
Ecoli	Random-sampling	69.348 ±	16.853	33.969 ±	9.853	103.317 ±	15.614
Ecoli	Stratified-sampling	65.678 ±	16.214	34.174 ±	10.710	99.852 ±	16.457
Ecoli	Balanced-sampling	83.869 ±	20.904	30.357 ±	7.087	114.226 ±	20.376
German	Windowing	315.252 ±	60.182	82.866 ±	5.220	398.118 ±	60.077
German	Cross-Validation	287.566 ±	54.049	83.857 ±	5.339	371.423 ±	53.413
German	Random-sampling	211.627 ±	51.692	83.245 ±	5.156	294.871 ±	51.783
German	Stratified-sampling	212.684 ±	54.545	83.006 ±	5.125	295.689 ±	53.830
German	Balanced-sampling	238.184 ±	51.813	84.412 ±	5.352	322.596 ±	51.356
Hypothyroid	Windowing	84.812 ±	19.108	28.291 ±	6.449	113.102 ±	20.727
Hypothyroid	Cross-Validation	122.317 ±	10.791	27.105 ±	6.877	149.422 ±	10.562
Hypothyroid	Random-sampling	15.667 ±	15.278	189.232 ±	110.454	204.899 ±	96.402
Hypothyroid	Stratified-sampling	30.645 ±	6.465	67.493 ±	22.683	98.138 ±	22.336
Hypothyroid	Balanced-sampling	45.353 ±	10.448	61.502 ±	18.798	106.854 ±	18.199
Kr-vs-kp	Windowing	198.034 ±	14.570	69.919 ±	4.871	267.953 ±	14.944
Kr-vs-kp	Cross-Validation	219.807 ±	16.870	69.345 ±	4.277	289.152 ±	17.014
Kr-vs-kp	Random-sampling	64.438 ±	18.816	98.961 ±	21.032	163.399 ±	21.636
Kr-vs-kp	Stratified-sampling	72.664 ±	18.341	92.724 ±	15.119	165.388 ±	15.947
Kr-vs-kp	Balanced-sampling	73.848 ±	18.721	91.842 ±	14.262	165.690 ±	15.840
Letter	Windowing	11862.644 ±	473.112	1248.697 ±	64.017	13111.341 ±	453.031
Letter	Cross-Validation	12431.372 ±	180.896	1165.793 ±	38.869	13597.165 ±	182.617
Letter	Random-sampling	7020.909 ±	385.222	1473.635 ±	81.356	8494.544 ±	358.576
Letter	Stratified-sampling	7102.767 ±	358.000	1461.702 ±	80.161	8564.469 ±	328.131
Letter	Balanced-sampling	7126.843 ±	381.507	1449.106 ±	76.567	8575.949 ±	354.232
Mushroom	Windowing	79.249 ±	7.033	76.881 ±	4.163	156.130 ±	7.189
Mushroom	Cross-Validation	77.237 ±	0.600	79.510 ±	1.744	156.747 ±	1.810
Mushroom	Random-sampling	18.228 ±	19.552	461.838 ±	353.124	480.066 ±	337.153
Mushroom	Stratified-sampling	31.126 ±	14.101	114.606 ±	23.525	145.732 ±	20.201

Continued on next page

Dataset	Method	L(H)		L(D | H)		MDL	
Mushroom	Balanced-sampling	31.879 ±	15.063	113.501 ±	22.427	145.380 ±	17.422
Segment	Windowing	348.723 ±	34.369	81.656 ±	10.719	430.379 ±	33.528
Segment	Cross-Validation	365.928 ±	22.569	79.045 ±	9.609	444.973 ±	22.295
Segment	Random-sampling	142.987 ±	22.538	135.754 ±	31.843	278.741 ±	31.578
Segment	Stratified-sampling	142.715 ±	18.438	126.640 ±	24.516	269.356 ±	26.762
Segment	Balanced-sampling	141.267 ±	17.852	127.325 ±	23.254	268.591 ±	26.010
Sick	Windowing	170.530 ±	26.600	50.476 ±	8.212	221.005 ±	26.977
Sick	Cross-Validation	182.701 ±	22.491	42.346 ±	7.910	225.047 ±	20.038
Sick	Random-sampling	21.786 ±	16.605	80.715 ±	38.277	102.501 ±	24.810
Sick	Stratified-sampling	31.126 ±	6.768	55.199 ±	13.736	86.325 ±	15.387
Sick	Balanced-sampling	57.996 ±	17.446	60.045 ±	9.531	118.040 ±	18.444
Splice	Windowing	725.951 ±	53.364	181.187 ±	11.871	907.139 ±	53.195
Splice	Cross-Validation	745.146 ±	51.142	179.689 ±	11.014	924.834 ±	52.532
Splice	Random-sampling	425.144 ±	52.153	187.097 ±	21.631	612.240 ±	47.209
Splice	Stratified-sampling	443.339 ±	51.337	188.061 ±	19.286	631.400 ±	48.312
Splice	Balanced-sampling	419.763 ±	41.676	188.473 ±	20.593	608.236 ±	40.687
Waveform-5000	Windowing	2418.668 ±	215.760	363.799 ±	56.499	2782.467 ±	224.433
Waveform-5000	Cross-Validation	2615.956 ±	94.305	415.810 ±	20.601	3031.766 ±	92.381
Waveform-5000	Random-sampling	1957.647 ±	203.398	413.447 ±	24.548	2371.094 ±	202.636
Waveform-5000	Stratified-sampling	1957.202 ±	199.174	417.104 ±	26.348	2374.306 ±	196.151
Waveform-5000	Balanced-sampling	1966.554 ±	193.650	417.152 ±	28.133	2383.706 ±	190.987

Table B.3: Predictive performance.

Dataset	Method	Test Acc		Test AUC	
adult	Windowing	86.355 ±	0.889	78.227 ±	1.161
adult	Cross-Validation	86.074 ±	0.390	77.080 ±	0.823
adult	Random-sampling	85.516 ±	0.423	76.131 ±	2.021
adult	Stratified-sampling	85.677 ±	0.401	76.680 ±	0.885
adult	Balanced-sampling	80.489 ±	0.722	81.956 ±	0.580
australian	Windowing	85.710 ±	4.355	85.471 ±	4.411
australian	Cross-Validation	86.536 ±	3.969	86.239 ±	4.041
australian	Random-sampling	85.101 ±	4.375	84.849 ±	4.517
australian	Stratified-sampling	85.391 ±	4.164	85.142 ±	4.266
australian	Balanced-sampling	85.536 ±	3.925	85.584 ±	3.854
breast	Windowing	94.829 ±	2.804	94.368 ±	3.117
breast	Cross-Validation	95.533 ±	2.674	95.058 ±	2.830
breast	Random-sampling	92.696 ±	3.821	91.687 ±	4.739
breast	Stratified-sampling	92.783 ±	3.485	91.956 ±	3.982
breast	Balanced-sampling	92.433 ±	3.558	92.301 ±	3.627
diabetes	Windowing	74.161 ±	4.864	70.041 ±	5.654
diabetes	Cross-Validation	74.756 ±	4.661	71.211 ±	5.027
diabetes	Random-sampling	72.280 ±	4.520	68.602 ±	5.403
diabetes	Stratified-sampling	73.222 ±	5.113	70.254 ±	5.721
diabetes	Balanced-sampling	71.018 ±	5.222	71.726 ±	4.937
ecoli	Windowing	82.777 ±	6.353	88.848 ±	4.134
ecoli	Cross-Validation	82.822 ±	5.467	88.873 ±	3.567
ecoli	Random-sampling	80.059 ±	6.268	86.924 ±	4.218
ecoli	Stratified-sampling	79.586 ±	6.227	86.721 ±	4.113
ecoli	Balanced-sampling	79.405 ±	6.360	86.981 ±	4.034
german	Windowing	71.660 ±	4.608	63.119 ±	5.518
german	Cross-Validation	71.300 ±	3.765	62.605 ±	4.388
german	Random-sampling	71.800 ±	3.782	62.867 ±	4.408
german	Stratified-sampling	71.640 ±	3.799	62.857 ±	4.546
german	Balanced-sampling	67.820 ±	4.448	66.833 ±	4.014
hypothyroid	Windowing	99.483 ±	0.346	98.880 ±	1.204
hypothyroid	Cross-Validation	99.528 ±	0.353	98.871 ±	1.259
hypothyroid	Random-sampling	94.340 ±	2.524	70.634 ±	23.378
hypothyroid	Stratified-sampling	96.877 ±	1.652	94.594 ±	4.769
hypothyroid	Balanced-sampling	96.236 ±	1.831	97.598 ±	1.421
kr-vs-kp	Windowing	99.302 ±	0.583	99.294 ±	0.594
kr-vs-kp	Cross-Validation	99.415 ±	0.433	99.412 ±	0.433
kr-vs-kp	Random-sampling	94.171 ±	2.959	94.139 ±	3.061
kr-vs-kp	Stratified-sampling	94.956 ±	1.766	94.956 ±	1.802
kr-vs-kp	Balanced-sampling	94.984 ±	1.727	94.996 ±	1.756
letter	Windowing	87.161 ±	2.074	93.324 ±	1.078
letter	Cross-Validation	87.943 ±	0.720	93.731 ±	0.375
letter	Random-sampling	82.216 ±	1.006	90.753 ±	0.523
letter	Stratified-sampling	82.376 ±	1.148	90.836 ±	0.597
letter	Balanced-sampling	82.430 ±	1.160	90.864 ±	0.603
mushroom	Windowing	100.000 ±	0.000	100.000 ±	0.000
mushroom	Cross-Validation	100.000 ±	0.000	100.000 ±	0.000
mushroom	Random-sampling	73.746 ±	23.610	73.625 ±	23.684
mushroom	Stratified-sampling	98.367 ±	0.813	98.312 ±	0.831

Continued on next page

Dataset	Method	Test Acc		Test AUC	
mushroom	Balanced-sampling	98.424 ±	0.819	98.376 ±	0.831
segment	Windowing	96.329 ±	1.655	97.859 ±	0.965
segment	Cross-Validation	96.710 ±	1.335	98.081 ±	0.779
segment	Random-sampling	90.719 ±	3.181	94.586 ±	1.855
segment	Stratified-sampling	91.515 ±	2.074	95.051 ±	1.210
segment	Balanced-sampling	91.455 ±	1.984	95.015 ±	1.157
sick	Windowing	98.688 ±	0.640	93.667 ±	3.370
sick	Cross-Validation	98.741 ±	0.523	93.662 ±	3.323
sick	Random-sampling	96.193 ±	1.887	75.662 ±	19.843
sick	Stratified-sampling	97.301 ±	1.051	86.908 ±	6.166
sick	Balanced-sampling	94.785 ±	1.855	94.812 ±	2.641
splice	Windowing	94.132 ±	1.682	95.626 ±	1.344
splice	Cross-Validation	94.216 ±	1.474	95.723 ±	1.125
splice	Random-sampling	89.997 ±	2.226	92.370 ±	1.951
splice	Stratified-sampling	90.339 ±	1.973	92.757 ±	1.572
splice	Balanced-sampling	89.846 ±	2.199	92.902 ±	1.570
waveform-5000	Windowing	83.802 ±	9.864	87.848 ±	7.402
waveform-5000	Cross-Validation	75.202 ±	1.989	81.396 ±	1.493
waveform-5000	Random-sampling	75.046 ±	2.159	81.279 ±	1.619
waveform-5000	Stratified-sampling	75.252 ±	1.981	81.431 ±	1.487
waveform-5000	Balanced-sampling	75.514 ±	2.143	81.628 ±	1.609

Experiment C results C

Table C.1: Evolution of sample properties over Windowing iterations.

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Adult	1	0	1099	0.375	0	0.047	0.701
Adult	1	1	9472	0.064	0.145	0.278	0.718
Adult	1	2	12825	0.078	0.132	0.351	0.720
Adult	1	3	13923	0.079	0.13	0.374	0.720
Adult	1	4	14320	0.081	0.13	0.382	0.720
Adult	1	5	14456	0.079	0.13	0.385	0.720
Adult	1	6	14522	0.081	0.129	0.386	0.720
Adult	2	0	1099	0.356	0	0.048	0.703
Adult	2	1	9341	0.062	0.145	0.276	0.719
Adult	2	2	12996	0.071	0.138	0.354	0.721
Adult	2	3	14083	0.075	0.135	0.377	0.721
Adult	2	4	14475	0.082	0.129	0.386	0.721
Adult	2	5	14686	0.081	0.129	0.39	0.721
Adult	2	6	14754	0.082	0.128	0.392	0.721
Adult	2	7	14791	0.081	0.129	0.392	0.720
Adult	2	8	14815	0.081	0.129	0.393	0.720
Adult	3	0	1099	0.372	0	0.048	0.701
Adult	3	1	9318	0.074	0.136	0.276	0.719
Adult	3	2	12902	0.081	0.129	0.352	0.721
Adult	3	3	13940	0.085	0.126	0.373	0.721
Adult	3	4	14331	0.092	0.121	0.381	0.721
Adult	3	5	14599	0.095	0.118	0.386	0.721
Adult	4	0	1099	0.368	0	0.048	0.701
Adult	4	1	9406	0.078	0.132	0.279	0.718
Adult	4	2	12924	0.072	0.136	0.355	0.721
Adult	4	3	13746	0.075	0.134	0.372	0.720
Adult	4	4	14229	0.081	0.129	0.382	0.721
Adult	5	0	1099	0.386	0.001	0.047	0.702
Adult	5	1	9289	0.068	0.141	0.275	0.718
Adult	5	2	12640	0.054	0.153	0.348	0.720
Adult	5	3	14112	0.075	0.134	0.379	0.720
Adult	5	4	14543	0.076	0.133	0.389	0.720
Adult	6	0	1099	0.348	0.001	0.048	0.703
Adult	6	1	9458	0.064	0.144	0.278	0.718
Adult	6	2	13017	0.082	0.128	0.353	0.721
Adult	6	3	14120	0.083	0.127	0.376	0.721
Adult	6	4	14609	0.079	0.131	0.386	0.721
Adult	6	5	14855	0.083	0.127	0.391	0.721
Adult	6	6	15019	0.085	0.125	0.394	0.721
Adult	6	7	15088	0.085	0.126	0.396	0.720
Adult	7	0	1099	0.372	0	0.047	0.700
Adult	7	1	9164	0.055	0.152	0.272	0.718
Adult	7	2	12732	0.064	0.144	0.348	0.720

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Adult	7	3	13732	0.074	0.136	0.368	0.720
Adult	7	4	14123	0.078	0.132	0.376	0.720
Adult	8	0	1099	0.362	0	0.047	0.703
Adult	8	1	9375	0.061	0.147	0.278	0.718
Adult	8	2	12906	0.078	0.132	0.354	0.720
Adult	8	3	14084	0.082	0.129	0.379	0.720
Adult	9	0	1099	0.362	0	0.047	0.701
Adult	9	1	9436	0.064	0.144	0.278	0.718
Adult	9	2	12945	0.069	0.139	0.353	0.720
Adult	9	3	13979	0.072	0.136	0.374	0.720
Adult	9	4	14369	0.076	0.133	0.381	0.720
Adult	9	5	14618	0.081	0.13	0.387	0.720
Adult	10	0	1099	0.379	0	0.048	0.703
Adult	10	1	9410	0.069	0.139	0.278	0.719
Adult	10	2	13037	0.079	0.131	0.358	0.721
Adult	10	3	13949	0.083	0.127	0.375	0.720
Adult	10	4	14449	0.086	0.124	0.386	0.720
Adult	10	5	14574	0.088	0.124	0.388	0.720
Australian	1	0	15	0.236	0.149	0.914	0.645
Australian	1	1	169	0.038	0.002	1	0.617
Australian	1	2	203	0.003	0.01	1	0.616
Australian	2	0	15	0.047	0.023	0.886	0.625
Australian	2	1	153	0.014	0.006	1	0.619
Australian	2	2	198	0.007	0.01	1	0.617
Australian	2	3	222	0.025	0.015	1	0.620
Australian	2	4	223	0.028	0.016	1	0.620
Australian	3	0	15	0.141	0.006	0.943	0.705
Australian	3	1	153	0.014	0.012	1	0.616
Australian	3	2	207	0.038	0.019	1	0.612
Australian	3	3	226	0.031	0.017	1	0.614
Australian	3	4	229	0.04	0.02	1	0.613
Australian	4	0	15	0.141	0.006	0.914	0.682
Australian	4	1	150	0.085	0.038	1	0.607
Australian	4	2	190	0.089	0.041	1	0.609
Australian	4	3	198	0.1	0.046	1	0.611
Australian	5	0	15	0.236	0.039	0.971	0.666
Australian	5	1	171	0.079	0	1	0.612
Australian	5	2	210	0.014	0.006	1	0.612
Australian	5	3	227	0.021	0.014	1	0.612
Australian	5	4	230	0.018	0.013	1	0.613
Australian	6	0	15	0.33	0.106	0.914	0.649
Australian	6	1	149	0.052	0.024	1	0.610
Australian	6	2	200	0.007	0.007	1	0.613
Australian	6	3	220	0.02	0.014	1	0.615
Australian	6	4	223	0.023	0.014	1	0.615
Australian	6	5	227	0.034	0.018	1	0.615
Australian	7	0	15	0.33	0.106	0.914	0.704
Australian	7	1	167	0.038	0.002	1	0.620
Australian	7	2	209	0.024	0.015	1	0.617
Australian	7	3	220	0.007	0.007	1	0.619
Australian	8	0	15	0.141	0.07	0.943	0.652

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Australian	8	1	148	0.066	0.03	1	0.617
Australian	8	2	202	0.028	0.016	1	0.610
Australian	8	3	219	0.016	0.013	1	0.614
Australian	9	0	15	0.141	0.07	0.914	0.690
Australian	9	1	170	0.034	0.003	0.971	0.602
Australian	9	2	208	0.014	0.006	1	0.613
Australian	10	0	15	0.047	0.023	0.914	0.664
Australian	10	1	160	0.052	0.001	1	0.625
Australian	10	2	204	0.014	0.006	1	0.619
Australian	10	3	223	0.028	0.016	1	0.619
Australian	10	4	235	0.045	0.022	1	0.619
Australian	10	5	236	0.042	0.021	1	0.619
Breast	1	0	15	0.236	0.001	0.968	0.813
Breast	1	1	74	0.134	0.009	1	0.602
Breast	1	2	92	0.061	0.034	1	0.590
Breast	2	0	15	0.424	0.088	0.935	0.753
Breast	2	1	73	0.088	0.023	1	0.601
Breast	2	2	93	0.038	0.045	1	0.592
Breast	2	3	105	0.034	0.047	1	0.593
Breast	2	4	109	0.02	0.055	1	0.589
Breast	3	0	15	0.236	0.001	1	0.796
Breast	3	1	75	0.103	0.018	1	0.599
Breast	3	2	107	0.033	0.088	1	0.595
Breast	3	3	119	0.018	0.077	1	0.594
Breast	3	4	120	0.011	0.073	1	0.594
Breast	3	5	127	0.04	0.092	1	0.594
Breast	4	0	15	0.33	0.024	0.935	0.711
Breast	4	1	66	0.021	0.08	1	0.596
Breast	4	2	78	0.018	0.078	1	0.593
Breast	4	3	89	0.024	0.081	1	0.594
Breast	4	4	101	0.007	0.062	1	0.592
Breast	4	5	118	0.011	0.074	1	0.594
Breast	4	6	121	0.017	0.077	1	0.594
Breast	4	7	122	0.023	0.081	1	0.595
Breast	5	0	15	0.236	0.001	0.839	0.852
Breast	5	1	75	0.01	0.06	1	0.600
Breast	5	2	105	0.02	0.079	1	0.595
Breast	5	3	122	0.035	0.089	1	0.595
Breast	5	4	129	0.017	0.076	1	0.596
Breast	5	5	136	0	0.066	1	0.595
Breast	5	6	138	0.01	0.072	1	0.594
Breast	6	0	15	0.141	0.008	0.903	0.703
Breast	6	1	61	0.127	0.011	1	0.582
Breast	6	2	84	0.017	0.056	1	0.581
Breast	6	3	96	0.044	0.096	1	0.581
Breast	6	4	104	0.014	0.075	1	0.578
Breast	6	5	108	0.027	0.083	1	0.579
Breast	7	0	15	0.236	0.001	0.935	0.801
Breast	7	1	75	0.066	0.032	1	0.610
Breast	7	2	96	0	0.066	1	0.602
Breast	8	0	15	0.236	0.001	0.935	0.780

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Breast	8	1	55	0.064	0.033	1	0.586
Breast	8	2	82	0.034	0.089	1	0.586
Breast	8	3	99	0.021	0.08	1	0.587
Breast	9	0	15	0.33	0.024	0.903	0.791
Breast	9	1	69	0.031	0.086	1	0.602
Breast	9	2	97	0.021	0.08	1	0.592
Breast	10	0	15	0.047	0.04	0.968	0.708
Breast	10	1	64	0.11	0.152	1	0.609
Breast	10	2	77	0.045	0.097	1	0.597
Breast	10	3	85	0.075	0.12	1	0.592
Breast	10	4	95	0.052	0.102	1	0.598
Diabetes	1	0	17	0.124	0.012	0.084	0.620
Diabetes	1	1	265	0.13	0.01	0.569	0.582
Diabetes	1	2	331	0.033	0.049	0.657	0.581
Diabetes	1	3	412	0.096	0.021	0.732	0.581
Diabetes	1	4	434	0.11	0.016	0.747	0.581
Diabetes	1	5	456	0.13	0.011	0.771	0.581
Diabetes	1	6	466	0.13	0.011	0.779	0.580
Diabetes	2	0	17	0.291	0.01	0.088	0.621
Diabetes	2	1	239	0.062	0.034	0.542	0.583
Diabetes	2	2	362	0.074	0.029	0.676	0.583
Diabetes	2	3	403	0.065	0.033	0.71	0.583
Diabetes	2	4	420	0.057	0.036	0.729	0.583
Diabetes	2	5	438	0.081	0.026	0.747	0.583
Diabetes	3	0	17	0.291	0.01	0.081	0.620
Diabetes	3	1	262	0.107	0.017	0.588	0.586
Diabetes	3	2	364	0.1	0.019	0.692	0.582
Diabetes	3	3	416	0.112	0.016	0.736	0.582
Diabetes	3	4	444	0.109	0.017	0.76	0.583
Diabetes	3	5	450	0.107	0.017	0.765	0.582
Diabetes	4	0	17	0.124	0.012	0.09	0.624
Diabetes	4	1	243	0.143	0.008	0.553	0.586
Diabetes	4	2	336	0.047	0.041	0.666	0.585
Diabetes	4	3	399	0.072	0.03	0.728	0.585
Diabetes	5	0	17	0.375	0.047	0.081	0.609
Diabetes	5	1	260	0.147	0.007	0.549	0.584
Diabetes	5	2	366	0.054	0.038	0.681	0.582
Diabetes	5	3	407	0.099	0.02	0.723	0.581
Diabetes	5	4	440	0.103	0.018	0.752	0.580
Diabetes	5	5	446	0.11	0.016	0.756	0.581
Diabetes	5	6	447	0.112	0.015	0.756	0.581
Diabetes	6	0	17	0.291	0.01	0.083	0.625
Diabetes	6	1	252	0.074	0.029	0.565	0.582
Diabetes	6	2	369	0.091	0.023	0.694	0.582
Diabetes	6	3	400	0.106	0.017	0.722	0.582
Diabetes	6	4	422	0.117	0.014	0.741	0.581
Diabetes	7	0	17	0.124	0.012	0.082	0.621
Diabetes	7	1	258	0.044	0.043	0.58	0.581
Diabetes	7	2	359	0.061	0.035	0.693	0.579
Diabetes	7	3	425	0.075	0.029	0.748	0.581
Diabetes	7	4	448	0.088	0.023	0.762	0.581

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Diabetes	7	5	455	0.089	0.023	0.771	0.581
Diabetes	7	6	460	0.095	0.021	0.774	0.581
Diabetes	8	0	17	0.041	0.044	0.085	0.628
Diabetes	8	1	272	0.037	0.046	0.575	0.582
Diabetes	8	2	370	0.058	0.036	0.678	0.580
Diabetes	8	3	412	0.055	0.037	0.714	0.580
Diabetes	9	0	17	0.291	0.01	0.082	0.621
Diabetes	9	1	266	0.011	0.06	0.572	0.584
Diabetes	9	2	380	0.112	0.016	0.693	0.580
Diabetes	9	3	416	0.099	0.02	0.728	0.580
Diabetes	9	4	432	0.102	0.019	0.745	0.580
Diabetes	9	5	435	0.106	0.018	0.748	0.580
Diabetes	10	0	17	0.124	0.012	0.084	0.636
Diabetes	10	1	257	0.129	0.011	0.58	0.582
Diabetes	10	2	364	0.035	0.047	0.693	0.581
Diabetes	10	3	419	0.093	0.022	0.747	0.581
Diabetes	10	4	437	0.086	0.024	0.76	0.580
Diabetes	10	5	449	0.093	0.022	0.771	0.581
Diabetes	10	6	466	0.098	0.02	0.787	0.581
Diabetes	10	7	475	0.103	0.018	0.798	0.580
Ecoli	1	0	7	0.218	1.922	0.105	0.828
Ecoli	1	1	90	0.095	0.197	0.642	0.925
Ecoli	1	2	115	0.104	0.134	0.752	0.922
Ecoli	1	3	124	0.107	0.158	0.768	0.920
Ecoli	2	0	7	0.071	0.552	0.111	0.924
Ecoli	2	1	93	0.109	0.194	0.685	0.923
Ecoli	2	2	125	0.109	0.218	0.749	0.919
Ecoli	3	0	7	0.078	0.477	0.111	0.959
Ecoli	3	1	96	0.104	0.279	0.66	0.923
Ecoli	3	2	115	0.106	0.22	0.72	0.920
Ecoli	4	0	7	0.214	1.024	0.105	0.906
Ecoli	4	1	98	0.108	0.109	0.682	0.922
Ecoli	4	2	117	0.11	0.101	0.749	0.919
Ecoli	4	3	127	0.114	0.116	0.765	0.918
Ecoli	5	0	7	0.33	1.356	0.102	0.811
Ecoli	5	1	87	0.109	0.189	0.652	0.922
Ecoli	5	2	112	0.11	0.178	0.725	0.918
Ecoli	6	0	7	0.128	1.448	0.108	0.953
Ecoli	6	1	90	0.108	0.15	0.66	0.924
Ecoli	6	2	113	0.104	0.163	0.733	0.921
Ecoli	6	3	125	0.106	0.181	0.768	0.920
Ecoli	6	4	129	0.105	0.163	0.774	0.919
Ecoli	6	5	129	0.105	0.163	0.774	0.919
Ecoli	7	0	7	0.218	1.967	0.1	0.829
Ecoli	7	1	93	0.102	0.15	0.65	0.924
Ecoli	7	2	129	0.102	0.17	0.741	0.920
Ecoli	8	0	7	0.137	0.727	0.108	0.918
Ecoli	8	1	89	0.098	0.205	0.639	0.924
Ecoli	8	2	121	0.11	0.256	0.741	0.918
Ecoli	8	3	126	0.111	0.232	0.763	0.918
Ecoli	8	4	126	0.111	0.232	0.763	0.918

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Ecoli	9	0	7	0.083	1.336	0.102	0.854
Ecoli	9	1	88	0.096	0.204	0.631	0.925
Ecoli	9	2	114	0.109	0.26	0.712	0.921
Ecoli	10	0	7	0.33	1.676	0.097	0.804
Ecoli	10	1	98	0.106	0.246	0.704	0.922
Ecoli	10	2	122	0.112	0.167	0.771	0.919
Ecoli	10	3	129	0.111	0.19	0.784	0.918
Ecoli	10	4	130	0.113	0.193	0.784	0.918
German	1	0	22	0.386	0.02	0.867	0.618
German	1	1	345	0.096	0.053	1	0.618
German	1	2	482	0.112	0.045	1	0.618
German	1	3	546	0.122	0.04	1	0.619
German	1	4	590	0.139	0.032	1	0.619
German	1	5	610	0.134	0.034	1	0.620
German	1	6	614	0.133	0.035	1	0.620
German	2	0	22	0.386	0.02	0.827	0.630
German	2	1	319	0.059	0.075	1	0.616
German	2	2	486	0.105	0.049	1	0.617
German	2	3	552	0.12	0.041	1	0.618
German	2	4	569	0.123	0.039	1	0.618
German	2	5	577	0.122	0.04	1	0.618
German	3	0	22	0.386	0.02	0.867	0.631
German	3	1	338	0.066	0.07	1	0.622
German	3	2	503	0.109	0.047	1	0.619
German	3	3	570	0.119	0.041	1	0.620
German	3	4	590	0.124	0.039	1	0.621
German	3	5	597	0.127	0.038	1	0.621
German	3	6	608	0.127	0.037	1	0.621
German	3	7	608	0.127	0.037	1	0.621
German	4	0	22	0.386	0.02	0.88	0.642
German	4	1	339	0.069	0.069	0.987	0.617
German	4	2	478	0.1	0.051	1	0.617
German	4	3	537	0.099	0.052	1	0.617
German	4	4	569	0.126	0.038	1	0.616
German	5	0	22	0.064	0.177	0.88	0.638
German	5	1	351	0.106	0.048	1	0.623
German	5	2	475	0.096	0.053	1	0.619
German	5	3	559	0.102	0.05	1	0.619
German	5	4	587	0.112	0.045	1	0.618
German	5	5	596	0.112	0.045	1	0.618
German	5	6	599	0.11	0.046	1	0.618
German	5	7	599	0.11	0.046	1	0.618
German	6	0	22	0.257	0.001	0.893	0.648
German	6	1	343	0.068	0.07	0.987	0.619
German	6	2	497	0.095	0.054	1	0.617
German	6	3	539	0.115	0.044	1	0.616
German	6	4	559	0.113	0.045	1	0.616
German	6	5	591	0.126	0.038	1	0.617
German	6	6	607	0.124	0.039	1	0.617
German	6	7	608	0.126	0.039	1	0.617
German	7	0	22	0.129	0.037	0.907	0.653

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
German	7	1	337	0.082	0.061	1	0.613
German	7	2	488	0.102	0.05	1	0.617
German	7	3	547	0.105	0.049	1	0.618
German	7	4	587	0.124	0.039	1	0.618
German	7	5	594	0.129	0.037	1	0.618
German	8	0	22	0.257	0.001	0.84	0.640
German	8	1	328	0.074	0.066	1	0.618
German	8	2	513	0.109	0.046	1	0.617
German	8	3	554	0.112	0.045	1	0.617
German	8	4	569	0.113	0.044	1	0.618
German	8	5	581	0.106	0.048	1	0.618
German	8	6	594	0.116	0.043	1	0.618
German	8	7	597	0.117	0.042	1	0.618
German	8	8	598	0.119	0.042	1	0.618
German	9	0	22	0.386	0.02	0.827	0.629
German	9	1	320	0.102	0.05	1	0.618
German	9	2	474	0.119	0.041	1	0.619
German	9	3	542	0.133	0.035	1	0.619
German	9	4	584	0.139	0.033	1	0.618
German	9	5	599	0.139	0.033	1	0.619
German	9	6	611	0.143	0.031	1	0.618
German	10	0	22	0.257	0.001	0.827	0.617
German	10	1	336	0.117	0.042	1	0.620
German	10	2	483	0.107	0.047	1	0.620
German	10	3	544	0.091	0.056	1	0.620
German	10	4	562	0.1	0.051	1	0.619
German	10	5	574	0.099	0.052	1	0.620
German	10	6	578	0.098	0.052	1	0.620
Hypothyroid	1	0	85	0.496	0.002	0.297	0.782
Hypothyroid	1	1	140	0.311	0.218	0.404	0.772
Hypothyroid	1	2	148	0.293	0.267	0.417	0.837
Hypothyroid	1	3	151	0.285	0.289	0.42	0.837
Hypothyroid	2	0	85	0.526	0.004	0.301	0.838
Hypothyroid	2	1	151	0.346	0.131	0.437	0.771
Hypothyroid	2	2	161	0.325	0.178	0.455	0.840
Hypothyroid	3	0	85	0.527	0.008	0.28	0.841
Hypothyroid	3	1	151	0.336	0.163	0.427	0.778
Hypothyroid	3	2	165	0.301	0.238	0.451	0.847
Hypothyroid	4	0	85	0.476	0.012	0.295	0.846
Hypothyroid	4	1	136	0.333	0.163	0.405	0.789
Hypothyroid	4	2	149	0.298	0.245	0.43	0.849
Hypothyroid	5	0	85	0.497	0.017	0.308	0.850
Hypothyroid	5	1	132	0.328	0.172	0.41	0.779
Hypothyroid	5	2	137	0.315	0.2	0.422	0.846
Hypothyroid	6	0	85	0.526	0.004	0.293	0.846
Hypothyroid	6	1	145	0.297	0.247	0.421	0.781
Hypothyroid	6	2	156	0.272	0.319	0.438	0.847
Hypothyroid	6	3	157	0.273	0.315	0.442	0.847
Hypothyroid	7	0	85	0.527	0.008	0.289	0.847
Hypothyroid	7	1	142	0.322	0.185	0.421	0.784
Hypothyroid	7	2	151	0.3	0.24	0.435	0.848

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Hypothyroid	8	0	85	0.476	0.017	0.289	0.849
Hypothyroid	8	1	123	0.326	0.183	0.382	0.783
Hypothyroid	8	2	136	0.292	0.264	0.404	0.848
Hypothyroid	8	3	147	0.27	0.326	0.423	0.848
Hypothyroid	8	4	147	0.27	0.326	0.423	0.849
Hypothyroid	9	0	85	0.506	0	0.3	0.849
Hypothyroid	9	1	142	0.305	0.236	0.418	0.782
Hypothyroid	9	2	150	0.29	0.271	0.424	0.846
Hypothyroid	10	0	85	0.516	0.008	0.285	0.847
Hypothyroid	10	1	153	0.326	0.176	0.44	0.846
Hypothyroid	10	2	167	0.296	0.25	0.455	0.847
Hypothyroid	10	3	167	0.296	0.25	0.455	0.847
Kr-vs-kp	1	0	72	0.079	0.018	0.947	0.690
Kr-vs-kp	1	1	242	0.071	0.015	1	0.716
Kr-vs-kp	1	2	258	0.076	0.017	1	0.716
Kr-vs-kp	1	3	259	0.074	0.016	1	0.716
Kr-vs-kp	2	0	72	0.059	0.001	0.947	0.696
Kr-vs-kp	2	1	169	0.054	0.011	0.987	0.707
Kr-vs-kp	2	2	191	0.055	0.011	1	0.713
Kr-vs-kp	3	0	72	0.059	0.001	0.96	0.722
Kr-vs-kp	3	1	205	0.065	0.014	1	0.719
Kr-vs-kp	3	2	225	0.054	0.01	1	0.717
Kr-vs-kp	3	3	226	0.049	0.01	1	0.718
Kr-vs-kp	3	4	227	0.052	0.01	1	0.717
Kr-vs-kp	4	0	72	0.098	0.007	0.947	0.695
Kr-vs-kp	4	1	230	0	0.001	1	0.721
Kr-vs-kp	4	2	238	0.006	0.002	1	0.719
Kr-vs-kp	5	0	72	0.04	0	0.947	0.694
Kr-vs-kp	5	1	244	0.028	0.005	0.987	0.700
Kr-vs-kp	5	2	268	0.027	0.005	0.987	0.700
Kr-vs-kp	5	3	268	0.027	0.005	0.987	0.700
Kr-vs-kp	6	0	72	0.098	0.007	0.973	0.707
Kr-vs-kp	6	1	184	0.107	0.028	1	0.713
Kr-vs-kp	6	2	205	0.058	0.012	1	0.713
Kr-vs-kp	6	3	207	0.058	0.012	1	0.713
Kr-vs-kp	7	0	72	0.079	0.003	0.947	0.692
Kr-vs-kp	7	1	209	0.078	0.003	1	0.712
Kr-vs-kp	7	2	226	0.031	0	1	0.715
Kr-vs-kp	7	3	226	0.031	0	1	0.715
Kr-vs-kp	8	0	72	0.157	0.023	0.947	0.701
Kr-vs-kp	8	1	222	0.013	0.001	1	0.713
Kr-vs-kp	8	2	233	0.003	0.002	1	0.713
Kr-vs-kp	8	3	238	0.006	0.002	1	0.713
Kr-vs-kp	8	4	238	0.006	0.002	1	0.713
Kr-vs-kp	9	0	72	0.117	0.011	0.947	0.702
Kr-vs-kp	9	1	252	0.089	0.005	1	0.714
Kr-vs-kp	9	2	265	0.057	0.001	1	0.713
Kr-vs-kp	10	0	72	0.079	0.018	0.947	0.716
Kr-vs-kp	10	1	224	0.107	0.028	1	0.714
Kr-vs-kp	10	2	238	0.124	0.036	1	0.714
Kr-vs-kp	10	3	240	0.13	0.038	1	0.714

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Letter	1	0	450	0.009	0.039	0.855	0.977
Letter	1	1	4962	0.008	0.037	0.968	0.974
Letter	1	2	6313	0.008	0.036	0.975	0.974
Letter	1	3	7054	0.008	0.036	0.979	0.974
Letter	1	4	7448	0.008	0.037	0.979	0.974
Letter	1	5	7703	0.008	0.037	0.979	0.974
Letter	1	6	7873	0.009	0.039	0.979	0.974
Letter	1	7	8051	0.009	0.038	0.986	0.974
Letter	1	8	8154	0.009	0.038	0.986	0.974
Letter	2	0	450	0.008	0.027	0.862	0.977
Letter	2	1	5076	0.009	0.04	0.989	0.974
Letter	2	2	6374	0.009	0.039	0.989	0.974
Letter	2	3	7113	0.008	0.035	0.993	0.974
Letter	2	4	7483	0.008	0.037	0.993	0.974
Letter	3	0	450	0.006	0.018	0.883	0.978
Letter	3	1	5031	0.008	0.033	0.986	0.974
Letter	3	2	6286	0.008	0.036	0.986	0.974
Letter	3	3	6999	0.008	0.035	0.986	0.974
Letter	3	4	7362	0.008	0.036	0.986	0.974
Letter	3	5	7663	0.008	0.037	0.986	0.974
Letter	3	6	7865	0.008	0.036	0.986	0.974
Letter	3	7	7940	0.008	0.036	0.993	0.974
Letter	4	0	450	0.008	0.028	0.855	0.977
Letter	4	1	4992	0.008	0.034	0.979	0.974
Letter	4	2	6231	0.009	0.038	0.986	0.974
Letter	4	3	6897	0.008	0.037	0.989	0.974
Letter	4	4	7177	0.009	0.04	0.989	0.974
Letter	5	0	450	0.009	0.051	0.865	0.977
Letter	5	1	4955	0.008	0.038	0.982	0.974
Letter	5	2	6349	0.009	0.039	0.982	0.974
Letter	5	3	7002	0.009	0.039	0.982	0.974
Letter	5	4	7367	0.009	0.039	0.982	0.974
Letter	5	5	7624	0.009	0.039	0.989	0.974
Letter	5	6	7813	0.009	0.04	0.989	0.974
Letter	5	7	7928	0.009	0.04	0.989	0.974
Letter	6	0	450	0.009	0.041	0.883	0.978
Letter	6	1	5014	0.008	0.034	0.979	0.974
Letter	6	2	6399	0.008	0.033	0.982	0.974
Letter	6	3	7105	0.008	0.034	0.982	0.974
Letter	6	4	7495	0.008	0.035	0.982	0.974
Letter	6	5	7726	0.008	0.035	0.986	0.974
Letter	6	6	7823	0.008	0.036	0.986	0.974
Letter	6	7	7903	0.008	0.037	0.986	0.974
Letter	7	0	450	0.008	0.029	0.869	0.977
Letter	7	1	5028	0.008	0.031	0.982	0.974
Letter	7	2	6294	0.008	0.032	0.986	0.974
Letter	7	3	6952	0.008	0.036	0.986	0.974
Letter	7	4	7287	0.008	0.037	0.989	0.974
Letter	8	0	450	0.008	0.036	0.894	0.978
Letter	8	1	4978	0.008	0.039	0.982	0.974
Letter	8	2	6226	0.008	0.038	0.986	0.974

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Letter	8	3	6962	0.008	0.037	0.989	0.974
Letter	8	4	7236	0.008	0.038	0.989	0.974
Letter	8	5	7478	0.009	0.039	0.989	0.974
Letter	8	6	7620	0.008	0.038	0.989	0.974
Letter	8	7	7665	0.009	0.039	0.989	0.974
Letter	9	0	450	0.009	0.04	0.883	0.978
Letter	9	1	5050	0.008	0.037	0.968	0.974
Letter	9	2	6428	0.008	0.035	0.968	0.974
Letter	9	3	7037	0.008	0.036	0.972	0.974
Letter	9	4	7425	0.008	0.037	0.975	0.974
Letter	9	5	7681	0.008	0.037	0.975	0.974
Letter	9	6	7861	0.008	0.036	0.975	0.974
Letter	10	0	450	0.008	0.027	0.869	0.977
Letter	10	1	5035	0.008	0.034	0.989	0.974
Letter	10	2	6360	0.008	0.036	0.989	0.974
Letter	10	3	6993	0.008	0.034	0.989	0.974
Mushroom	1	0	183	0.058	0.002	0.899	0.698
Mushroom	1	1	223	0.079	0.004	0.983	0.690
Mushroom	2	0	183	0.051	0.001	0.874	0.682
Mushroom	2	1	221	0.023	0	0.975	0.679
Mushroom	3	0	183	0.074	0.014	0.916	0.696
Mushroom	3	1	219	0.048	0.008	0.983	0.692
Mushroom	4	0	183	0.051	0.008	0.916	0.676
Mushroom	4	1	205	0.086	0.018	0.966	0.676
Mushroom	5	0	183	0.051	0.008	0.899	0.682
Mushroom	5	1	208	0.102	0.024	0.941	0.675
Mushroom	6	0	183	0.02	0.003	0.924	0.692
Mushroom	6	1	231	0.04	0	1	0.681
Mushroom	7	0	183	0.065	0.002	0.95	0.710
Mushroom	7	1	225	0.098	0.008	0.966	0.691
Mushroom	8	0	183	0.027	0	0.857	0.683
Mushroom	8	1	202	0	0.001	0.941	0.686
Mushroom	9	0	183	0.042	0	0.866	0.696
Mushroom	9	1	215	0.01	0	0.966	0.695
Mushroom	10	0	183	0.042	0	0.84	0.689
Mushroom	10	1	209	0.024	0	0.933	0.688
Segment	1	0	52	0.055	0.081	0.061	0.926
Segment	1	1	269	0.103	0.377	0.225	0.903
Segment	1	2	311	0.108	0.395	0.249	0.902
Segment	1	3	344	0.104	0.367	0.264	0.902
Segment	1	4	365	0.108	0.402	0.273	0.901
Segment	1	5	372	0.108	0.407	0.277	0.900
Segment	1	6	376	0.109	0.414	0.278	0.900
Segment	1	7	391	0.113	0.432	0.284	0.900
Segment	1	8	398	0.114	0.441	0.286	0.900
Segment	2	0	52	0.029	0.028	0.061	0.928
Segment	2	1	243	0.097	0.352	0.212	0.904
Segment	2	2	301	0.105	0.412	0.238	0.902
Segment	2	3	333	0.108	0.426	0.251	0.901
Segment	3	0	52	0.074	0.181	0.061	0.923
Segment	3	1	254	0.083	0.235	0.217	0.904

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Segment	3	2	319	0.092	0.293	0.252	0.902
Segment	4	0	52	0.031	0.031	0.061	0.928
Segment	4	1	289	0.099	0.33	0.243	0.902
Segment	4	2	343	0.106	0.376	0.27	0.901
Segment	4	3	374	0.108	0.404	0.284	0.900
Segment	4	4	383	0.112	0.422	0.287	0.900
Segment	5	0	52	0.05	0.07	0.058	0.931
Segment	5	1	234	0.108	0.387	0.196	0.905
Segment	5	2	293	0.117	0.479	0.228	0.902
Segment	5	3	327	0.121	0.514	0.248	0.900
Segment	6	0	52	0.049	0.101	0.06	0.926
Segment	6	1	273	0.095	0.336	0.232	0.902
Segment	6	2	329	0.104	0.413	0.259	0.900
Segment	6	3	342	0.107	0.437	0.265	0.900
Segment	6	4	356	0.108	0.456	0.27	0.900
Segment	6	5	360	0.107	0.436	0.273	0.900
Segment	7	0	52	0.033	0.042	0.061	0.927
Segment	7	1	261	0.099	0.309	0.22	0.903
Segment	7	2	339	0.108	0.395	0.265	0.901
Segment	7	3	360	0.112	0.433	0.273	0.900
Segment	8	0	52	0.047	0.076	0.058	0.928
Segment	8	1	265	0.104	0.399	0.221	0.902
Segment	8	2	315	0.107	0.416	0.247	0.901
Segment	9	0	52	0.057	0.098	0.062	0.926
Segment	9	1	280	0.099	0.419	0.233	0.902
Segment	9	2	355	0.097	0.385	0.273	0.901
Segment	9	3	370	0.1	0.392	0.279	0.901
Segment	9	4	374	0.101	0.399	0.282	0.900
Segment	10	0	52	0.048	0.098	0.061	0.927
Segment	10	1	277	0.103	0.409	0.228	0.902
Segment	10	2	327	0.108	0.455	0.253	0.900
Sick	1	0	85	0.591	0.005	0.301	0.656
Sick	1	1	221	0.304	0.233	0.52	0.676
Sick	1	2	269	0.291	0.247	0.57	0.673
Sick	1	3	280	0.283	0.257	0.581	0.672
Sick	1	4	287	0.293	0.245	0.586	0.671
Sick	2	0	85	0.607	0.001	0.291	0.664
Sick	2	1	196	0.317	0.218	0.487	0.672
Sick	2	2	249	0.327	0.208	0.535	0.674
Sick	2	3	265	0.328	0.206	0.554	0.673
Sick	3	0	85	0.658	0.012	0.285	0.665
Sick	3	1	212	0.314	0.222	0.514	0.675
Sick	3	2	241	0.303	0.235	0.543	0.673
Sick	4	0	85	0.607	0.001	0.304	0.666
Sick	4	1	201	0.348	0.185	0.505	0.674
Sick	4	2	237	0.307	0.229	0.542	0.673
Sick	4	3	251	0.296	0.242	0.555	0.673
Sick	4	4	257	0.288	0.25	0.563	0.672
Sick	4	5	258	0.29	0.248	0.564	0.672
Sick	5	0	85	0.658	0.012	0.297	0.662
Sick	5	1	183	0.344	0.19	0.471	0.678

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Sick	5	2	229	0.305	0.231	0.527	0.676
Sick	5	3	258	0.29	0.248	0.554	0.676
Sick	5	4	261	0.296	0.243	0.557	0.676
Sick	6	0	85	0.658	0.012	0.315	0.657
Sick	6	1	204	0.263	0.28	0.505	0.671
Sick	6	2	244	0.273	0.269	0.544	0.672
Sick	7	0	85	0.641	0.003	0.293	0.647
Sick	7	1	222	0.402	0.133	0.514	0.666
Sick	7	2	249	0.361	0.173	0.546	0.671
Sick	7	3	269	0.355	0.178	0.568	0.671
Sick	8	0	85	0.54	0.026	0.298	0.664
Sick	8	1	197	0.355	0.178	0.507	0.661
Sick	8	2	246	0.298	0.239	0.56	0.667
Sick	8	3	268	0.269	0.273	0.577	0.667
Sick	9	0	85	0.54	0.026	0.301	0.664
Sick	9	1	193	0.378	0.156	0.501	0.671
Sick	9	2	239	0.363	0.169	0.551	0.672
Sick	9	3	252	0.337	0.197	0.567	0.672
Sick	9	4	254	0.339	0.194	0.571	0.672
Sick	9	5	256	0.342	0.191	0.572	0.672
Sick	10	0	85	0.624	0	0.303	0.657
Sick	10	1	195	0.315	0.22	0.509	0.673
Sick	10	2	221	0.266	0.277	0.543	0.676
Sick	10	3	232	0.274	0.267	0.558	0.676
Sick	10	4	237	0.277	0.263	0.564	0.676
Splice	1	0	72	0.217	0.012	0.838	0.692
Splice	1	1	601	0.08	0.027	0.986	0.714
Splice	1	2	739	0.065	0.038	0.986	0.715
Splice	1	3	803	0.064	0.042	0.986	0.715
Splice	1	4	853	0.072	0.034	0.986	0.714
Splice	2	0	72	0.182	0.004	0.838	0.693
Splice	2	1	593	0.068	0.034	0.979	0.713
Splice	2	2	727	0.075	0.03	0.979	0.713
Splice	2	3	772	0.065	0.037	0.983	0.714
Splice	2	4	787	0.062	0.039	0.983	0.714
Splice	3	0	72	0.087	0.025	0.838	0.699
Splice	3	1	594	0.08	0.028	0.841	0.684
Splice	3	2	747	0.075	0.035	0.979	0.713
Splice	3	3	781	0.076	0.034	0.979	0.713
Splice	3	4	782	0.075	0.035	0.979	0.713
Splice	4	0	72	0.21	0.026	0.838	0.693
Splice	4	1	565	0.102	0.015	0.855	0.686
Splice	4	2	733	0.084	0.027	0.855	0.687
Splice	4	3	790	0.08	0.028	0.855	0.687
Splice	4	4	803	0.076	0.031	0.855	0.687
Splice	5	0	72	0.157	0	0.841	0.695
Splice	5	1	608	0.095	0.017	0.983	0.713
Splice	5	2	814	0.094	0.018	0.986	0.714
Splice	5	3	859	0.086	0.023	0.986	0.714
Splice	5	4	867	0.084	0.025	0.986	0.714
Splice	6	0	72	0.229	0.022	0.841	0.692

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Splice	6	1	580	0.107	0.011	0.986	0.713
Splice	6	2	746	0.091	0.02	0.986	0.714
Splice	6	3	795	0.089	0.021	0.986	0.714
Splice	6	4	824	0.087	0.024	0.986	0.714
Splice	6	5	837	0.09	0.022	0.986	0.714
Splice	6	6	838	0.091	0.022	0.986	0.714
Splice	7	0	72	0.139	0.031	0.838	0.696
Splice	7	1	628	0.105	0.016	0.986	0.714
Splice	7	2	781	0.083	0.028	0.986	0.715
Splice	7	3	838	0.08	0.032	0.986	0.715
Splice	7	4	850	0.08	0.034	0.986	0.715
Splice	8	0	72	0.207	0.017	0.838	0.694
Splice	8	1	600	0.102	0.015	0.979	0.712
Splice	8	2	745	0.073	0.032	0.983	0.714
Splice	8	3	790	0.061	0.042	0.983	0.714
Splice	9	0	72	0.169	0	0.845	0.696
Splice	9	1	563	0.101	0.014	0.983	0.713
Splice	9	2	737	0.084	0.029	0.983	0.713
Splice	9	3	811	0.07	0.036	0.983	0.714
Splice	9	4	824	0.066	0.041	0.983	0.714
Splice	9	5	831	0.064	0.043	0.983	0.714
Splice	9	6	833	0.063	0.044	0.983	0.714
Splice	10	0	72	0.144	0.001	0.838	0.695
Splice	10	1	574	0.096	0.016	0.983	0.713
Splice	10	2	761	0.082	0.029	0.986	0.714
Splice	10	3	815	0.081	0.031	0.986	0.714
Splice	10	4	836	0.077	0.037	0.986	0.715
Splice	10	5	841	0.077	0.038	0.986	0.715
Splice	10	6	842	0.077	0.038	0.986	0.715
Waveform-5000	1	0	112	0.044	0.009	0.157	0.741
Waveform-5000	1	1	1477	0.018	0.001	0.788	0.701
Waveform-5000	1	2	2192	0.015	0	0.872	0.696
Waveform-5000	1	3	2600	0.009	0	0.905	0.695
Waveform-5000	1	4	2895	0.01	0	0.923	0.694
Waveform-5000	1	5	3057	0.008	0	0.932	0.694
Waveform-5000	1	6	3177	0.007	0	0.938	0.694
Waveform-5000	1	7	3275	0.007	0	0.943	0.693
Waveform-5000	1	8	3346	0.006	0	0.946	0.693
Waveform-5000	1	9	3409	0.006	0	0.949	0.693
Waveform-5000	1	10	3453	0.005	0	0.951	0.693
Waveform-5000	1	11	3487	0.005	0	0.952	0.693
Waveform-5000	1	12	3510	0.004	0	0.953	0.693
Waveform-5000	1	13	3524	0.004	0	0.953	0.693
Waveform-5000	1	14	3534	0.005	0	0.954	0.693
Waveform-5000	2	0	112	0.045	0.009	0.159	0.741
Waveform-5000	2	1	1462	0.003	0	0.785	0.701
Waveform-5000	2	2	2211	0.012	0.001	0.872	0.696
Waveform-5000	2	3	2626	0.016	0	0.902	0.695
Waveform-5000	2	4	2893	0.009	0	0.919	0.694
Waveform-5000	2	5	3057	0.008	0	0.928	0.694
Waveform-5000	2	6	3161	0.007	0	0.934	0.694

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Waveform-5000	2	7	3246	0.007	0	0.937	0.693
Waveform-5000	2	8	3331	0.006	0	0.941	0.693
Waveform-5000	2	9	3391	0.004	0	0.944	0.693
Waveform-5000	2	10	3429	0.003	0	0.946	0.693
Waveform-5000	2	11	3455	0.002	0	0.947	0.693
Waveform-5000	3	0	112	0.023	0.001	0.156	0.741
Waveform-5000	3	1	1403	0.03	0.003	0.777	0.701
Waveform-5000	3	2	2198	0.024	0.002	0.873	0.696
Waveform-5000	3	3	2635	0.019	0.001	0.906	0.695
Waveform-5000	3	4	2901	0.012	0	0.922	0.694
Waveform-5000	3	5	3061	0.015	0	0.93	0.694
Waveform-5000	3	6	3181	0.015	0	0.936	0.693
Waveform-5000	3	7	3281	0.012	0	0.942	0.693
Waveform-5000	3	8	3344	0.012	0	0.945	0.693
Waveform-5000	3	9	3398	0.01	0	0.948	0.693
Waveform-5000	3	10	3439	0.012	0	0.949	0.693
Waveform-5000	3	11	3471	0.012	0	0.95	0.693
Waveform-5000	3	12	3521	0.01	0	0.952	0.693
Waveform-5000	3	13	3548	0.007	0	0.953	0.693
Waveform-5000	3	14	3560	0.007	0	0.954	0.693
Waveform-5000	4	0	112	0.027	0.003	0.157	0.741
Waveform-5000	4	1	1423	0.009	0	0.782	0.702
Waveform-5000	4	2	2144	0.012	0	0.869	0.697
Waveform-5000	4	3	2571	0.012	0	0.902	0.695
Waveform-5000	4	4	2865	0.011	0	0.921	0.694
Waveform-5000	4	5	3067	0.009	0	0.932	0.694
Waveform-5000	4	6	3176	0.01	0	0.938	0.694
Waveform-5000	4	7	3270	0.008	0	0.941	0.693
Waveform-5000	4	8	3351	0.007	0	0.944	0.693
Waveform-5000	4	9	3400	0.004	0	0.946	0.693
Waveform-5000	4	10	3447	0.003	0	0.948	0.693
Waveform-5000	4	11	3492	0.004	0	0.95	0.693
Waveform-5000	4	12	3516	0.004	0	0.951	0.693
Waveform-5000	4	13	3530	0.004	0	0.951	0.693
Waveform-5000	4	14	3534	0.004	0	0.951	0.693
Waveform-5000	5	0	112	0.049	0.01	0.157	0.741
Waveform-5000	5	1	1435	0.021	0.001	0.782	0.701
Waveform-5000	5	2	2183	0.006	0	0.874	0.697
Waveform-5000	5	3	2601	0.009	0	0.905	0.695
Waveform-5000	5	4	2869	0.009	0	0.921	0.694
Waveform-5000	5	5	3042	0.006	0	0.93	0.694
Waveform-5000	5	6	3144	0.005	0	0.936	0.694
Waveform-5000	5	7	3235	0.005	0	0.941	0.694
Waveform-5000	5	8	3312	0.004	0	0.945	0.693
Waveform-5000	5	9	3369	0.004	0	0.947	0.693
Waveform-5000	6	0	112	0.094	0.036	0.157	0.741
Waveform-5000	6	1	1487	0.034	0.005	0.788	0.701
Waveform-5000	6	2	2212	0.02	0.001	0.873	0.696
Waveform-5000	6	3	2668	0.01	0	0.908	0.695
Waveform-5000	6	4	2921	0.006	0	0.923	0.694
Waveform-5000	6	5	3100	0.008	0	0.931	0.694

Continued on next page

Dataset	CV fold	Iteration	Instances	S.D. C.D.	KL div.	Sim1	Red
Waveform-5000	6	6	3228	0.006	0	0.938	0.693
Waveform-5000	7	0	112	0.034	0.005	0.157	0.742
Waveform-5000	7	1	1433	0.024	0.002	0.779	0.701
Waveform-5000	7	2	2169	0.018	0.001	0.869	0.696
Waveform-5000	7	3	2597	0.015	0	0.903	0.695
Waveform-5000	7	4	2865	0.01	0	0.921	0.694
Waveform-5000	7	5	3050	0.006	0	0.932	0.694
Waveform-5000	7	6	3154	0.005	0	0.938	0.694
Waveform-5000	7	7	3258	0.005	0	0.943	0.693
Waveform-5000	7	8	3343	0.006	0	0.946	0.693
Waveform-5000	7	9	3418	0.006	0	0.95	0.693
Waveform-5000	8	0	112	0.021	0.002	0.158	0.741
Waveform-5000	8	1	1434	0.039	0.005	0.781	0.701
Waveform-5000	8	2	2144	0.017	0.001	0.872	0.697
Waveform-5000	8	3	2573	0.016	0.001	0.905	0.695
Waveform-5000	8	4	2819	0.017	0.001	0.922	0.694
Waveform-5000	8	5	3019	0.012	0	0.933	0.694
Waveform-5000	8	6	3144	0.012	0	0.939	0.694
Waveform-5000	8	7	3237	0.012	0	0.943	0.693
Waveform-5000	9	0	112	0.058	0.015	0.159	0.741
Waveform-5000	9	1	1365	0.025	0.002	0.773	0.702
Waveform-5000	9	2	2120	0.012	0	0.868	0.697
Waveform-5000	9	3	2550	0.011	0	0.903	0.695
Waveform-5000	9	4	2820	0.007	0	0.919	0.694
Waveform-5000	9	5	2979	0.006	0	0.927	0.694
Waveform-5000	9	6	3120	0.005	0	0.935	0.694
Waveform-5000	10	0	112	0.054	0.011	0.157	0.741
Waveform-5000	10	1	1448	0.024	0.002	0.784	0.701
Waveform-5000	10	2	2170	0.018	0.001	0.871	0.696
Waveform-5000	10	3	2639	0.01	0	0.908	0.695
Waveform-5000	10	4	2898	0.01	0	0.923	0.694
Waveform-5000	10	5	3066	0.009	0	0.931	0.694

Table C.2: Evolution of the MDL and the predictive performance over Windowing iterations.

Dataset	CV fold	Iteration	L(H)	L(D | H)	MDL	Test Acc	Test Auc
Adult	1	0	46.04	2909.53	2955.57	83.81	72.06
Adult	1	1	1534.61	2501.58	4036.19	80.86	75.75
Adult	1	2	1770.68	2419.78	4190.46	84.69	77.51
Adult	1	3	1468.57	2404.72	3873.29	84.91	76.10
Adult	1	4	1070.28	2376.45	3446.73	85.96	77.43
Adult	1	5	1194.31	2386.52	3580.84	86.12	77.16
Adult	1	6	1355.62	2356.30	3711.92	86.76	77.69
Adult	2	0	124.50	2471.96	2596.46	84.83	79.01
Adult	2	1	1604.22	2385.55	3989.77	80.80	75.92
Adult	2	2	2489.36	2421.54	4910.90	84.38	77.01
Adult	2	3	1867.49	2327.18	4194.67	86.04	79.01
Adult	2	4	1753.03	2351.37	4104.40	86.16	79.56
Adult	2	5	1008.59	2320.18	3328.76	86.76	80.07
Adult	2	6	637.21	2361.45	2998.66	86.33	77.67
Adult	2	7	1277.58	2297.79	3575.37	87.47	80.13
Adult	2	8	1250.70	2316.20	3566.90	87.37	80.00
Adult	3	0	59.65	2682.00	2741.65	83.62	70.74
Adult	3	1	1080.31	2620.09	3700.41	80.38	75.32
Adult	3	2	1070.28	2514.76	3585.03	83.64	71.98
Adult	3	3	1137.51	2504.26	3641.77	84.91	76.19
Adult	3	4	1488.92	2475.66	3964.58	85.26	75.92
Adult	3	5	874.97	2501.59	3376.57	85.32	75.58
Adult	4	0	37.23	3005.49	3042.72	84.11	72.09
Adult	4	1	2365.75	2385.18	4750.93	82.17	75.44
Adult	4	2	1350.00	2425.69	3775.69	84.54	76.79
Adult	4	3	1518.99	2435.55	3954.54	85.28	77.04
Adult	4	4	1293.19	2404.24	3697.43	85.97	77.47
Adult	5	0	46.04	2635.43	2681.47	83.82	73.69
Adult	5	1	2319.86	2497.75	4817.60	79.12	71.59
Adult	5	2	1927.14	2481.43	4408.57	84.44	78.69
Adult	5	3	2142.52	2474.53	4617.04	84.83	77.66
Adult	5	4	1848.68	2439.07	4287.75	85.83	78.38
Adult	6	0	162.88	2533.87	2696.76	83.85	78.39
Adult	6	1	1937.60	2410.85	4348.45	79.57	74.52
Adult	6	2	1154.70	2416.73	3571.42	85.50	77.19
Adult	6	3	2227.18	2328.19	4555.37	85.77	79.36
Adult	6	4	2813.69	2470.10	5283.79	85.01	78.04
Adult	6	5	1598.22	2335.65	3933.88	86.51	79.00
Adult	6	6	1714.11	2376.16	4090.27	86.08	78.45
Adult	6	7	1048.24	2367.70	3415.94	86.81	78.82
Adult	7	0	179.73	2607.86	2787.59	83.95	72.42
Adult	7	1	1485.76	2536.79	4022.55	81.10	76.32
Adult	7	2	1188.77	2516.98	3705.75	82.68	77.80
Adult	7	3	1912.37	2417.58	4329.95	86.14	78.14
Adult	7	4	1228.81	2432.38	3661.19	86.53	77.89
Adult	8	0	32.42	2970.97	3003.39	84.15	73.03
Adult	8	1	1440.11	2519.89	3960.01	82.13	78.92
Adult	8	2	1365.34	2486.09	3851.43	83.50	75.56
Adult	8	3	3980.44	2384.00	6364.44	85.48	78.10

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Adult	9	0	59.65	2490.40	2550.05	85.14	74.62
Adult	9	1	2469.09	2320.89	4789.97	81.70	76.79
Adult	9	2	1805.56	2265.80	4071.37	85.97	78.13
Adult	9	3	1067.05	2294.84	3361.89	86.88	79.22
Adult	9	4	2381.36	2311.04	4692.40	86.57	79.29
Adult	9	5	1374.46	2216.94	3591.40	87.29	79.23
Adult	10	0	58.84	3019.01	3077.85	83.82	73.28
Adult	10	1	2282.48	2406.27	4688.75	82.82	76.12
Adult	10	2	2175.36	2419.60	4594.96	83.80	76.16
Adult	10	3	2988.04	2323.70	5311.74	84.66	79.35
Adult	10	4	1152.74	2332.96	3485.69	86.90	79.09
Adult	10	5	1101.89	2308.21	3410.10	86.40	78.36
Australian	1	0	17.61	71.85	89.47	50.72	53.72
Australian	1	1	50.84	33.84	84.68	85.51	85.26
Australian	1	2	69.27	29.75	99.01	89.86	89.49
Australian	2	0	8.81	55.10	63.91	78.26	79.23
Australian	2	1	48.84	52.97	101.82	82.61	83.08
Australian	2	2	78.07	52.63	130.71	78.26	77.69
Australian	2	3	69.27	50.88	120.15	85.51	85.26
Australian	2	4	82.88	49.39	132.27	82.61	83.08
Australian	3	0	8.81	44.19	53.00	85.51	86.41
Australian	3	1	67.27	43.96	111.23	84.06	84.36
Australian	3	2	109.30	43.10	152.40	82.61	83.08
Australian	3	3	101.30	41.48	142.78	84.06	83.97
Australian	3	4	89.69	41.73	131.42	82.61	81.92
Australian	4	0	8.81	60.87	69.68	73.91	74.53
Australian	4	1	91.69	54.70	146.39	81.16	80.52
Australian	4	2	46.84	60.05	106.90	75.36	74.07
Australian	4	3	94.50	60.44	154.93	72.46	71.43
Australian	5	0	8.81	60.22	69.03	73.91	71.56
Australian	5	1	67.27	36.56	103.82	82.61	80.94
Australian	5	2	84.88	38.20	123.09	86.96	87.27
Australian	5	3	62.46	32.29	94.75	88.41	87.69
Australian	5	4	69.27	32.77	102.04	91.30	90.62
Australian	6	0	8.81	36.15	44.96	88.41	89.47
Australian	6	1	76.07	49.40	125.48	79.71	81.28
Australian	6	2	91.69	31.61	123.30	88.41	87.69
Australian	6	3	69.27	32.99	102.26	89.86	90.49
Australian	6	4	89.69	31.47	121.16	88.41	88.88
Australian	6	5	69.27	36.55	105.82	85.51	85.65
Australian	7	0	8.81	42.33	51.14	86.96	87.56
Australian	7	1	116.92	37.09	154.01	84.06	83.74
Australian	7	2	78.07	40.28	118.36	84.06	83.74
Australian	7	3	82.88	40.07	122.95	81.16	80.81
Australian	8	0	8.81	36.15	44.96	88.41	89.47
Australian	8	1	17.61	36.15	53.77	88.41	89.47
Australian	8	2	100.50	46.86	147.36	84.06	84.34
Australian	8	3	78.07	32.43	110.51	88.41	89.18
Australian	9	0	8.81	48.33	57.14	84.06	84.34
Australian	9	1	86.88	44.82	131.70	79.71	78.31
Australian	9	2	93.69	44.77	138.46	85.51	83.87

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Australian	10	0	22.42	65.58	88.00	68.12	67.19
Australian	10	1	17.61	44.96	62.57	85.51	85.36
Australian	10	2	53.65	42.78	96.43	85.51	85.65
Australian	10	3	96.50	44.41	140.91	85.51	85.06
Australian	10	4	76.07	43.04	119.12	85.51	85.65
Australian	10	5	46.84	43.63	90.47	86.96	86.97
Breast	1	0	10.17	37.09	47.26	85.51	82.08
Breast	1	1	30.51	37.24	67.75	89.86	89.31
Breast	1	2	42.85	35.20	78.05	89.86	89.31
Breast	2	0	12.17	23.90	36.07	89.86	85.42
Breast	2	1	30.51	22.09	52.60	95.65	94.72
Breast	2	2	34.68	21.21	55.89	98.55	98.89
Breast	2	3	57.02	22.49	79.51	92.75	91.53
Breast	2	4	57.02	22.49	79.51	92.75	91.53
Breast	3	0	10.17	27.36	37.53	89.86	85.42
Breast	3	1	42.85	29.12	71.97	94.20	92.64
Breast	3	2	20.34	25.74	46.08	92.75	94.44
Breast	3	3	46.85	21.66	68.51	100.00	100.00
Breast	3	4	28.51	25.51	54.02	94.20	94.58
Breast	3	5	46.85	21.66	68.51	100.00	100.00
Breast	4	0	10.17	43.59	53.76	83.82	76.09
Breast	4	1	38.68	25.55	64.23	92.65	93.38
Breast	4	2	28.51	28.28	56.79	91.18	88.02
Breast	4	3	28.51	30.93	59.44	91.18	89.08
Breast	4	4	38.68	30.20	68.88	94.12	93.43
Breast	4	5	69.36	27.62	96.98	97.06	96.71
Breast	4	6	40.68	27.66	68.34	92.65	90.19
Breast	4	7	48.85	27.66	76.51	92.65	90.19
Breast	5	0	12.17	9.53	21.70	91.18	87.50
Breast	5	1	51.02	30.78	81.80	89.71	90.15
Breast	5	2	28.51	19.88	48.39	95.59	96.59
Breast	5	3	30.51	20.03	50.53	94.12	94.51
Breast	5	4	34.68	19.87	54.55	98.53	98.86
Breast	5	5	55.02	23.70	78.72	95.59	95.64
Breast	5	6	55.02	18.05	73.07	95.59	96.59
Breast	6	0	10.17	47.24	57.41	80.88	79.55
Breast	6	1	28.51	22.84	51.35	94.12	93.56
Breast	6	2	57.02	25.22	82.24	97.06	95.83
Breast	6	3	38.68	22.12	60.80	97.06	97.73
Breast	6	4	48.85	20.80	69.64	97.06	96.78
Breast	6	5	42.85	31.50	74.35	92.65	93.37
Breast	7	0	12.17	25.65	37.82	94.12	95.45
Breast	7	1	28.51	25.62	54.13	80.88	80.49
Breast	7	2	59.02	23.16	82.18	98.53	98.86
Breast	8	0	12.17	29.99	42.16	91.18	90.34
Breast	8	1	30.51	29.33	59.84	91.18	90.34
Breast	8	2	18.34	22.83	41.17	92.65	94.32
Breast	8	3	38.68	17.61	56.29	100.00	100.00
Breast	9	0	12.17	30.27	42.44	91.18	88.45
Breast	9	1	20.34	32.27	52.61	86.76	88.83
Breast	9	2	46.85	21.45	68.30	97.06	97.73

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Breast	10	0	12.17	42.34	54.51	82.35	76.89
Breast	10	1	40.68	39.58	80.26	89.71	87.31
Breast	10	2	46.85	41.28	88.13	88.24	86.17
Breast	10	3	53.02	41.01	94.03	82.35	80.68
Breast	10	4	48.85	39.58	88.43	91.18	89.39
Diabetes	1	0	20.00	71.17	91.17	70.13	66.78
Diabetes	1	1	14.00	63.13	77.13	76.62	66.67
Diabetes	1	2	44.00	65.98	109.98	76.62	75.19
Diabetes	1	3	38.00	61.66	99.66	75.32	75.89
Diabetes	1	4	146.00	62.55	208.55	74.03	74.04
Diabetes	1	5	62.00	68.90	130.90	70.13	71.89
Diabetes	1	6	86.00	60.87	146.87	72.73	74.74
Diabetes	2	0	8.00	76.52	84.52	58.44	51.81
Diabetes	2	1	104.00	65.47	169.47	71.43	62.67
Diabetes	2	2	74.00	67.13	141.13	72.73	73.04
Diabetes	2	3	92.00	71.75	163.75	70.13	62.52
Diabetes	2	4	38.00	63.20	101.20	71.43	69.48
Diabetes	2	5	50.00	71.17	121.17	67.53	63.07
Diabetes	3	0	14.00	76.76	90.76	62.34	54.81
Diabetes	3	1	20.00	75.00	95.00	57.14	59.33
Diabetes	3	2	80.00	73.92	153.92	62.34	65.04
Diabetes	3	3	74.00	73.83	147.83	66.23	64.63
Diabetes	3	4	38.00	70.96	108.96	71.43	67.78
Diabetes	3	5	74.00	66.54	140.54	71.43	67.78
Diabetes	4	0	20.00	71.07	91.07	67.53	69.89
Diabetes	4	1	50.00	76.33	126.33	66.23	54.41
Diabetes	4	2	68.00	75.87	143.87	59.74	59.63
Diabetes	4	3	38.00	69.70	107.70	62.34	63.33
Diabetes	5	0	14.00	56.81	70.81	77.92	80.44
Diabetes	5	1	14.00	68.28	82.28	74.03	65.52
Diabetes	5	2	74.00	71.02	145.02	67.53	69.89
Diabetes	5	3	26.00	67.84	93.84	76.62	71.78
Diabetes	5	4	50.00	70.54	120.54	70.13	66.78
Diabetes	5	5	110.00	64.29	174.29	71.43	67.78
Diabetes	5	6	74.00	67.14	141.14	68.83	63.22
Diabetes	6	0	8.00	63.61	71.61	77.92	71.93
Diabetes	6	1	14.00	69.05	83.05	74.03	66.37
Diabetes	6	2	56.00	71.40	127.40	66.23	67.19
Diabetes	6	3	38.00	65.63	103.63	74.03	74.89
Diabetes	6	4	44.00	65.63	109.63	74.03	74.89
Diabetes	7	0	14.00	56.92	70.92	58.44	50.11
Diabetes	7	1	8.00	70.80	78.80	53.25	63.15
Diabetes	7	2	8.00	68.04	76.04	74.03	64.67
Diabetes	7	3	26.00	56.77	82.77	80.52	81.59
Diabetes	7	4	62.00	59.30	121.30	75.32	68.22
Diabetes	7	5	50.00	58.88	108.88	81.82	79.19
Diabetes	7	6	62.00	58.40	120.40	77.92	70.22
Diabetes	8	0	20.00	76.13	96.13	55.84	48.96
Diabetes	8	1	50.00	66.37	116.37	67.53	63.93
Diabetes	8	2	8.00	64.91	72.91	76.62	68.37
Diabetes	8	3	38.00	65.86	103.86	76.62	69.22

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Diabetes	9	0	8.00	72.15	80.15	68.42	59.38
Diabetes	9	1	14.00	73.42	87.42	52.63	59.38
Diabetes	9	2	38.00	71.01	109.01	68.42	67.69
Diabetes	9	3	86.00	68.68	154.68	67.11	63.92
Diabetes	9	4	50.00	70.07	120.07	69.74	70.54
Diabetes	9	5	44.00	69.48	113.48	69.74	69.62
Diabetes	10	0	14.00	66.20	80.20	73.68	71.69
Diabetes	10	1	8.00	65.64	73.64	75.00	64.38
Diabetes	10	2	38.00	66.61	104.61	73.68	69.85
Diabetes	10	3	38.00	60.55	98.55	78.95	72.92
Diabetes	10	4	80.00	65.90	145.90	72.37	65.15
Diabetes	10	5	140.00	64.33	204.33	71.05	64.15
Diabetes	10	6	194.00	53.43	247.43	77.63	72.85
Diabetes	10	7	68.00	60.17	128.17	78.95	70.15
Ecoli	1	0	11.81	102.69	114.50	47.06	62.82
Ecoli	1	1	105.50	22.25	127.75	82.35	89.85
Ecoli	1	2	97.69	17.78	115.47	91.18	94.58
Ecoli	1	3	89.88	18.56	108.44	88.24	93.01
Ecoli	2	0	11.81	73.81	85.61	52.94	68.08
Ecoli	2	1	50.84	25.63	76.47	85.29	89.18
Ecoli	2	2	175.76	27.64	203.40	73.53	80.76
Ecoli	3	0	19.61	87.94	107.55	17.65	51.05
Ecoli	3	1	97.69	51.49	149.17	67.65	78.63
Ecoli	3	2	74.27	47.53	121.80	82.35	86.68
Ecoli	4	0	11.81	114.42	126.23	38.24	61.47
Ecoli	4	1	128.92	33.93	162.85	79.41	87.61
Ecoli	4	2	136.73	30.38	167.10	82.35	89.17
Ecoli	4	3	152.34	30.38	182.72	82.35	89.17
Ecoli	5	0	11.81	76.34	88.14	64.71	72.88
Ecoli	5	1	121.11	22.81	143.92	73.53	83.36
Ecoli	5	2	82.07	28.92	110.99	76.47	85.08
Ecoli	6	0	11.81	86.55	98.36	50.00	63.27
Ecoli	6	1	35.23	38.88	74.11	61.76	77.34
Ecoli	6	2	66.46	33.44	99.90	82.35	88.85
Ecoli	6	3	97.69	37.28	134.96	82.35	88.28
Ecoli	6	4	74.27	33.60	107.87	88.24	91.73
Ecoli	6	5	74.27	33.60	107.87	88.24	91.73
Ecoli	7	0	11.81	100.39	112.20	45.45	60.84
Ecoli	7	1	74.27	41.41	115.68	75.76	84.12
Ecoli	7	2	89.88	34.62	124.50	87.88	91.37
Ecoli	8	0	19.61	86.66	106.27	45.45	64.44
Ecoli	8	1	82.07	35.70	117.78	63.64	77.52
Ecoli	8	2	136.73	32.03	168.76	66.67	78.15
Ecoli	8	3	121.11	34.83	155.94	72.73	82.20
Ecoli	8	4	121.11	34.83	155.94	72.73	82.20
Ecoli	9	0	19.61	62.39	82.01	69.70	77.95
Ecoli	9	1	97.69	23.79	121.48	90.91	93.76
Ecoli	9	2	136.73	22.21	158.93	84.85	90.60
Ecoli	10	0	11.81	99.64	111.45	45.45	56.42
Ecoli	10	1	97.69	30.34	128.02	75.76	86.43
Ecoli	10	2	121.11	28.08	149.19	69.70	81.13

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D | H)	MDL	Test Acc	Test Auc
Ecoli	10	3	82.07	22.91	104.99	87.88	93.00
Ecoli	10	4	105.50	20.95	126.45	87.88	92.67
German	1	0	2.00	91.26	93.26	70.00	50.00
German	1	1	265.08	89.64	354.72	63.00	64.05
German	1	2	284.37	80.74	365.11	71.00	63.10
German	1	3	307.69	91.54	399.23	63.00	55.48
German	1	4	368.27	84.08	452.35	74.00	66.19
German	1	5	305.05	85.46	390.50	72.00	63.81
German	1	6	312.37	83.87	396.24	72.00	63.81
German	2	0	9.32	88.18	97.50	70.00	63.33
German	2	1	278.37	94.46	372.83	57.00	52.14
German	2	2	342.27	86.46	428.73	67.00	57.38
German	2	3	300.34	86.22	386.56	70.00	63.33
German	2	4	362.98	86.39	449.37	71.00	62.14
German	2	5	347.66	86.71	434.37	74.00	65.24
German	3	0	2.00	91.26	93.26	70.00	50.00
German	3	1	261.05	78.29	339.34	65.00	65.48
German	3	2	305.01	74.76	379.77	70.00	60.48
German	3	3	210.44	93.72	304.16	68.00	61.90
German	3	4	186.47	74.77	261.24	73.00	67.38
German	3	5	113.86	84.79	198.65	72.00	62.86
German	3	6	278.37	73.60	351.97	72.00	66.67
German	3	7	278.37	73.60	351.97	72.00	66.67
German	4	0	2.00	91.26	93.26	70.00	50.00
German	4	1	188.47	87.20	275.68	68.00	63.81
German	4	2	259.73	83.19	342.92	77.00	67.38
German	4	3	321.01	87.78	408.80	74.00	66.19
German	4	4	299.69	88.73	388.42	73.00	61.67
German	5	0	41.97	75.66	117.63	55.00	51.67
German	5	1	201.79	81.78	283.58	73.00	64.52
German	5	2	340.98	85.50	426.48	66.00	58.57
German	5	3	213.79	83.21	297.01	68.00	60.00
German	5	4	385.59	85.62	471.21	73.00	65.48
German	5	5	252.40	81.08	333.49	70.00	60.48
German	5	6	263.05	81.50	344.55	72.00	63.81
German	5	7	263.05	81.50	344.55	72.00	63.81
German	6	0	11.32	87.82	99.14	73.00	57.86
German	6	1	263.73	92.05	355.78	57.00	54.05
German	6	2	307.66	83.29	390.95	69.00	68.33
German	6	3	266.37	79.99	346.36	73.00	67.38
German	6	4	390.95	81.95	472.90	70.00	69.05
German	6	5	235.08	82.55	317.63	68.00	63.81
German	6	6	257.73	83.06	340.78	69.00	67.38
German	6	7	257.73	83.06	340.78	69.00	67.38
German	7	0	27.97	94.69	122.65	66.00	55.71
German	7	1	220.44	96.37	316.81	57.00	50.24
German	7	2	312.34	84.34	396.68	70.00	67.14
German	7	3	344.30	87.36	431.67	66.00	55.71
German	7	4	279.69	74.62	354.31	71.00	59.29
German	7	5	341.62	71.68	413.31	69.00	57.86
German	8	0	29.97	90.28	120.25	62.00	55.71

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
German	8	1	199.79	78.07	277.86	67.00	64.05
German	8	2	275.05	83.51	358.55	61.00	55.95
German	8	3	344.30	84.02	428.32	66.00	57.62
German	8	4	265.05	88.15	353.20	67.00	58.33
German	8	5	338.30	85.39	423.70	63.00	57.38
German	8	6	293.69	84.77	378.47	67.00	58.33
German	8	7	297.01	81.12	378.13	71.00	63.10
German	8	8	297.01	81.12	378.13	71.00	63.10
German	9	0	11.32	83.10	94.42	69.00	55.00
German	9	1	175.15	95.30	270.45	61.00	55.95
German	9	2	307.01	88.26	395.27	68.00	54.29
German	9	3	358.30	85.59	443.89	73.00	61.67
German	9	4	290.37	89.25	379.62	69.00	58.81
German	9	5	312.34	89.44	401.78	70.00	59.52
German	9	6	296.34	89.76	386.10	69.00	57.86
German	10	0	2.00	91.26	93.26	70.00	50.00
German	10	1	139.86	91.28	231.14	68.00	57.14
German	10	2	334.34	81.73	416.06	69.00	59.76
German	10	3	372.95	81.82	454.76	66.00	58.57
German	10	4	220.44	84.80	305.23	72.00	62.86
German	10	5	220.44	84.06	304.50	74.00	64.29
German	10	6	172.47	89.88	262.35	72.00	60.95
Hypothyroid	1	0	20.72	81.77	102.48	96.56	96.62
Hypothyroid	1	1	65.01	39.09	104.10	98.15	95.87
Hypothyroid	1	2	82.72	38.62	121.34	98.94	96.28
Hypothyroid	1	3	91.58	38.62	130.20	98.94	96.28
Hypothyroid	2	0	20.72	100.43	121.15	95.24	95.98
Hypothyroid	2	1	65.01	32.55	97.56	99.47	98.20
Hypothyroid	2	2	100.58	35.92	136.50	99.47	98.19
Hypothyroid	3	0	20.72	100.28	121.00	91.25	63.70
Hypothyroid	3	1	73.86	35.44	109.31	98.67	94.55
Hypothyroid	3	2	56.15	29.33	85.48	99.47	99.72
Hypothyroid	4	0	38.57	80.50	119.07	92.57	89.77
Hypothyroid	4	1	82.86	31.05	113.91	98.94	96.28
Hypothyroid	4	2	91.72	42.60	134.33	97.88	90.96
Hypothyroid	5	0	20.72	77.49	98.20	95.23	76.89
Hypothyroid	5	1	73.86	38.09	111.96	98.41	97.59
Hypothyroid	5	2	73.86	31.33	105.20	99.20	99.58
Hypothyroid	6	0	20.72	81.90	102.62	94.69	97.22
Hypothyroid	6	1	109.44	29.96	139.40	97.61	90.85
Hypothyroid	6	2	65.01	22.13	87.13	100.00	100.00
Hypothyroid	6	3	82.72	22.13	104.85	100.00	100.00
Hypothyroid	7	0	20.72	70.84	91.55	96.82	95.15
Hypothyroid	7	1	82.72	26.35	109.07	99.47	98.14
Hypothyroid	7	2	56.15	26.68	82.82	99.20	96.42
Hypothyroid	8	0	20.72	54.49	75.21	98.14	99.02
Hypothyroid	8	1	65.01	22.19	87.20	99.73	99.86
Hypothyroid	8	2	56.15	35.42	91.56	98.94	97.86
Hypothyroid	8	3	91.58	22.19	113.77	99.73	99.86
Hypothyroid	8	4	91.58	22.19	113.77	99.73	99.86
Hypothyroid	9	0	20.72	57.10	77.82	98.14	99.02

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Hypothyroid	9	1	56.15	18.73	74.88	100.00	100.00
Hypothyroid	9	2	82.72	18.73	101.46	100.00	100.00
Hypothyroid	10	0	11.86	107.97	119.82	94.96	86.25
Hypothyroid	10	1	91.72	20.07	111.80	99.73	98.28
Hypothyroid	10	2	91.58	23.47	115.05	99.73	99.86
Hypothyroid	10	3	91.58	23.47	115.05	99.73	99.86
Kr-vs-kp	1	0	34.68	111.19	145.87	94.38	94.31
Kr-vs-kp	1	1	167.40	69.06	236.46	99.38	99.37
Kr-vs-kp	1	2	210.25	66.35	276.60	99.69	99.67
Kr-vs-kp	1	3	210.25	66.35	276.60	99.69	99.67
Kr-vs-kp	2	0	34.68	117.49	152.17	93.75	93.60
Kr-vs-kp	2	1	200.08	86.86	286.93	97.81	97.77
Kr-vs-kp	2	2	200.08	81.66	281.74	98.13	98.09
Kr-vs-kp	3	0	42.85	132.25	175.10	90.94	90.74
Kr-vs-kp	3	1	159.23	64.95	224.18	99.38	99.35
Kr-vs-kp	3	2	208.25	65.14	273.39	99.69	99.70
Kr-vs-kp	3	3	208.25	63.14	271.39	100.00	100.00
Kr-vs-kp	3	4	216.42	63.14	279.56	100.00	100.00
Kr-vs-kp	4	0	75.53	114.40	189.93	79.06	79.17
Kr-vs-kp	4	1	159.23	72.76	231.99	98.44	98.42
Kr-vs-kp	4	2	175.57	67.74	243.31	99.69	99.67
Kr-vs-kp	5	0	59.19	131.10	190.29	89.69	89.22
Kr-vs-kp	5	1	208.25	98.32	306.57	96.56	96.62
Kr-vs-kp	5	2	240.93	76.53	317.46	99.06	99.02
Kr-vs-kp	5	3	240.93	76.53	317.46	99.06	99.02
Kr-vs-kp	6	0	34.68	111.27	145.95	94.06	93.93
Kr-vs-kp	6	1	232.76	133.37	366.13	86.56	86.88
Kr-vs-kp	6	2	200.08	68.84	268.92	99.69	99.70
Kr-vs-kp	6	3	208.25	68.84	277.09	99.69	99.70
Kr-vs-kp	7	0	42.85	115.84	158.69	89.66	89.78
Kr-vs-kp	7	1	183.74	85.88	269.62	97.49	97.46
Kr-vs-kp	7	2	183.74	83.18	266.92	98.12	98.12
Kr-vs-kp	7	3	183.74	83.18	266.92	98.12	98.12
Kr-vs-kp	8	0	26.51	105.68	132.19	93.73	93.48
Kr-vs-kp	8	1	226.59	67.85	294.44	99.06	99.07
Kr-vs-kp	8	2	216.42	62.10	278.52	99.37	99.37
Kr-vs-kp	8	3	218.42	60.59	279.01	99.69	99.70
Kr-vs-kp	8	4	218.42	60.59	279.01	99.69	99.70
Kr-vs-kp	9	0	42.85	93.82	136.67	94.67	94.64
Kr-vs-kp	9	1	183.74	65.71	249.45	98.75	98.68
Kr-vs-kp	9	2	240.93	65.66	306.59	100.00	100.00
Kr-vs-kp	10	0	42.85	96.38	139.22	94.36	94.43
Kr-vs-kp	10	1	175.57	73.50	249.07	98.12	98.03
Kr-vs-kp	10	2	224.59	63.74	288.33	99.37	99.34
Kr-vs-kp	10	3	183.74	59.60	243.34	100.00	100.00
Letter	1	0	904.54	5817.78	6722.31	55.00	76.60
Letter	1	1	9133.18	1396.94	10530.11	82.35	90.82
Letter	1	2	10213.92	1363.56	11577.48	84.15	91.76
Letter	1	3	11476.57	1219.77	12696.35	87.30	93.40
Letter	1	4	11829.69	1214.86	13044.54	87.30	93.40
Letter	1	5	12000.89	1246.59	13247.48	86.60	93.03

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D | H)	MDL	Test Acc	Test Auc
Letter	1	6	11915.29	1323.70	13238.99	86.20	92.82
Letter	1	7	11936.69	1214.17	13150.86	87.05	93.27
Letter	1	8	12118.60	1220.68	13339.28	87.80	93.66
Letter	2	0	1000.84	5605.99	6606.83	53.60	75.87
Letter	2	1	9507.69	1452.72	10960.41	82.65	90.98
Letter	2	2	10909.45	1269.25	12178.70	85.35	92.38
Letter	2	3	11540.77	1236.85	12777.63	87.70	93.60
Letter	2	4	12086.50	1230.02	13316.52	87.85	93.68
Letter	3	0	958.04	5601.01	6559.05	54.50	76.35
Letter	3	1	9143.88	1356.71	10500.58	82.00	90.64
Letter	3	2	11198.36	1378.26	12576.62	83.10	91.21
Letter	3	3	11166.26	1292.11	12458.37	86.80	93.14
Letter	3	4	11829.69	1210.82	13040.51	87.40	93.45
Letter	3	5	12097.20	1262.54	13359.73	88.05	93.79
Letter	3	6	12086.50	1190.20	13276.70	87.80	93.66
Letter	3	7	12247.00	1191.22	13438.22	88.70	94.12
Letter	4	0	1032.94	5833.09	6866.03	53.75	75.95
Letter	4	1	8619.55	1423.60	10043.16	80.35	89.78
Letter	4	2	11091.36	1239.76	12331.11	85.90	92.67
Letter	4	3	11936.69	1138.00	13074.69	87.95	93.73
Letter	4	4	12268.40	1198.95	13467.36	87.00	93.24
Letter	5	0	925.94	5952.38	6878.32	53.80	75.98
Letter	5	1	9186.68	1393.75	10580.43	81.95	90.62
Letter	5	2	9914.31	1302.16	11216.46	84.95	92.17
Letter	5	3	11016.45	1268.80	12285.25	85.80	92.62
Letter	5	4	11615.68	1258.90	12874.58	87.15	93.32
Letter	5	5	12043.70	1256.80	13300.50	88.10	93.81
Letter	5	6	11925.99	1235.49	13161.48	86.55	93.01
Letter	5	7	12493.11	1228.54	13721.65	88.05	93.79
Letter	6	0	958.04	5697.60	6655.64	54.05	76.11
Letter	6	1	9422.09	1453.51	10875.60	81.60	90.43
Letter	6	2	10995.05	1279.96	12275.01	85.95	92.69
Letter	6	3	12257.70	1198.26	13455.97	85.00	92.20
Letter	6	4	12000.89	1202.15	13203.05	87.85	93.68
Letter	6	5	12236.30	1153.79	13390.10	89.10	94.33
Letter	6	6	12493.11	1200.32	13693.43	88.40	93.97
Letter	6	7	12439.61	1199.76	13639.37	88.40	93.97
Letter	7	0	968.74	5488.44	6457.18	57.05	77.67
Letter	7	1	9315.08	1463.45	10778.53	81.75	90.51
Letter	7	2	10920.15	1353.83	12273.98	84.00	91.68
Letter	7	3	11840.39	1227.75	13068.13	86.70	93.09
Letter	7	4	11829.69	1239.74	13069.43	88.10	93.81
Letter	8	0	1022.24	5517.59	6539.83	53.95	76.06
Letter	8	1	9272.28	1462.09	10734.37	80.25	89.73
Letter	8	2	10962.95	1361.43	12324.38	83.80	91.58
Letter	8	3	11583.58	1270.61	12854.19	85.70	92.56
Letter	8	4	11722.68	1283.94	13006.63	87.45	93.47
Letter	8	5	12268.40	1206.55	13474.96	87.35	93.42
Letter	8	6	12354.01	1191.76	13545.77	87.60	93.55
Letter	8	7	12482.41	1232.22	13714.64	87.50	93.50
Letter	9	0	958.04	5746.89	6704.92	52.10	75.10

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D | H)	MDL	Test Acc	Test Auc
Letter	9	1	9357.88	1395.54	10753.43	81.60	90.43
Letter	9	2	11273.26	1275.17	12548.43	86.35	92.90
Letter	9	3	11744.08	1213.02	12957.11	86.50	92.98
Letter	9	4	11829.69	1168.10	12997.79	87.15	93.32
Letter	9	5	12075.80	1260.66	13336.46	86.95	93.21
Letter	9	6	12086.50	1150.69	13237.19	87.80	93.66
Letter	10	0	979.44	5848.86	6828.30	51.95	75.01
Letter	10	1	9454.19	1395.71	10849.89	81.95	90.62
Letter	10	2	10834.55	1268.40	12102.95	83.45	91.39
Letter	10	3	11733.38	1260.48	12993.86	85.85	92.64
Mushroom	1	0	54.38	90.29	144.67	99.38	99.36
Mushroom	1	1	90.76	74.29	165.04	100.00	100.00
Mushroom	2	0	54.38	84.47	138.85	99.02	98.98
Mushroom	2	1	71.30	78.68	149.98	100.00	100.00
Mushroom	3	0	23.46	107.52	130.98	98.89	98.85
Mushroom	3	1	71.30	76.07	147.36	100.00	100.00
Mushroom	4	0	23.46	140.12	163.58	98.03	97.96
Mushroom	4	1	87.30	75.36	162.65	100.00	100.00
Mushroom	5	0	23.46	151.84	175.30	97.66	97.57
Mushroom	5	1	87.30	78.55	165.85	100.00	100.00
Mushroom	6	0	23.46	97.30	120.76	99.14	99.10
Mushroom	6	1	78.76	85.28	164.04	100.00	100.00
Mushroom	7	0	23.46	120.93	144.39	98.52	98.47
Mushroom	7	1	71.30	76.72	148.01	100.00	100.00
Mushroom	8	0	23.46	143.60	167.06	97.41	97.31
Mushroom	8	1	90.76	72.55	163.31	100.00	100.00
Mushroom	9	0	54.38	89.99	144.37	98.89	98.85
Mushroom	9	1	78.76	73.94	152.70	100.00	100.00
Mushroom	10	0	23.46	110.44	133.90	98.03	97.96
Mushroom	10	1	84.76	76.18	160.93	100.00	100.00
Segment	1	0	67.19	331.59	398.78	77.92	87.12
Segment	1	1	329.80	88.16	417.96	93.94	96.46
Segment	1	2	284.52	86.25	370.77	93.94	96.46
Segment	1	3	329.80	80.06	409.86	94.37	96.72
Segment	1	4	356.96	82.49	439.46	94.81	96.97
Segment	1	5	366.02	77.86	443.87	95.24	97.22
Segment	1	6	311.69	73.96	385.65	97.84	98.74
Segment	1	7	366.02	74.35	440.37	96.54	97.98
Segment	1	8	356.96	77.97	434.93	96.54	97.98
Segment	2	0	58.14	377.48	435.62	75.76	85.86
Segment	2	1	257.36	94.22	351.57	94.81	96.97
Segment	2	2	302.63	76.95	379.58	96.54	97.98
Segment	2	3	320.74	81.98	402.72	96.54	97.98
Segment	3	0	67.19	334.71	401.91	77.92	87.12
Segment	3	1	257.36	129.91	387.26	91.77	95.20
Segment	3	2	266.41	83.03	349.44	93.07	95.96
Segment	4	0	58.14	232.61	290.75	81.39	89.14
Segment	4	1	275.47	74.58	350.05	94.37	96.72
Segment	4	2	329.80	81.08	410.87	93.94	96.46
Segment	4	3	320.74	79.76	400.50	96.10	97.73
Segment	4	4	338.85	74.46	413.32	97.84	98.74

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Segment	5	0	67.19	279.63	346.82	77.06	86.62
Segment	5	1	257.36	98.70	356.06	93.94	96.46
Segment	5	2	329.80	75.14	404.94	95.24	97.22
Segment	5	3	384.13	79.92	464.05	95.24	97.22
Segment	6	0	67.19	391.80	458.99	73.59	84.60
Segment	6	1	266.41	79.57	345.98	95.67	97.47
Segment	6	2	320.74	86.47	407.22	96.97	98.23
Segment	6	3	356.96	86.40	443.37	97.40	98.48
Segment	6	4	320.74	90.31	411.05	96.97	98.23
Segment	6	5	293.58	90.70	384.27	95.67	97.47
Segment	7	0	67.19	260.42	327.61	80.52	88.64
Segment	7	1	257.36	92.79	350.15	93.94	96.46
Segment	7	2	347.91	91.98	439.89	95.24	97.22
Segment	7	3	338.85	81.82	420.67	96.97	98.23
Segment	8	0	76.25	324.40	400.65	74.89	85.35
Segment	8	1	284.52	87.90	372.42	94.37	96.72
Segment	8	2	302.63	93.59	396.22	94.81	96.97
Segment	9	0	67.19	212.61	279.80	80.52	88.64
Segment	9	1	230.19	101.50	331.69	87.45	92.68
Segment	9	2	293.58	86.97	380.55	95.67	97.47
Segment	9	3	347.91	88.89	436.80	95.24	97.22
Segment	9	4	329.80	91.94	421.73	95.67	97.47
Segment	10	0	58.14	302.28	360.42	78.79	87.63
Segment	10	1	293.58	83.75	377.33	95.67	97.47
Segment	10	2	347.91	74.50	422.41	96.54	97.98
Sick	1	0	23.72	45.80	69.52	98.68	91.16
Sick	1	1	124.01	44.40	168.41	98.68	97.26
Sick	1	2	125.87	54.70	180.57	98.68	91.16
Sick	1	3	163.30	64.66	227.96	96.83	88.14
Sick	1	4	141.59	64.21	205.79	98.15	88.85
Sick	2	0	17.72	122.10	139.81	91.27	68.15
Sick	2	1	88.44	46.45	134.88	98.41	91.38
Sick	2	2	147.59	46.42	194.01	98.15	89.30
Sick	2	3	147.59	47.41	194.99	98.68	89.58
Sick	3	0	2.00	129.97	131.97	93.90	50.00
Sick	3	1	149.44	51.56	201.00	97.61	84.50
Sick	3	2	141.59	43.90	185.49	99.20	95.51
Sick	4	0	31.57	58.83	90.40	97.61	84.50
Sick	4	1	116.15	57.63	173.78	98.14	90.88
Sick	4	2	163.30	52.32	215.62	98.14	94.95
Sick	4	3	157.30	50.96	208.27	97.61	96.70
Sick	4	4	180.88	52.70	233.58	97.88	92.77
Sick	4	5	173.02	52.70	225.72	97.88	92.77
Sick	5	0	2.00	129.97	131.97	93.90	50.00
Sick	5	1	92.58	50.53	143.11	96.82	90.17
Sick	5	2	145.73	47.65	193.37	98.14	90.88
Sick	5	3	165.16	42.72	207.88	99.47	95.65
Sick	5	4	165.16	43.31	208.47	99.20	93.48
Sick	6	0	23.72	82.68	106.40	96.55	79.87
Sick	6	1	161.44	49.67	211.11	97.88	92.77
Sick	6	2	210.45	38.98	249.43	99.73	97.83

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Sick	7	0	31.57	57.19	88.77	95.76	75.38
Sick	7	1	70.86	47.89	118.75	98.14	86.82
Sick	7	2	124.01	49.35	173.36	98.94	95.37
Sick	7	3	163.30	46.77	210.07	99.20	95.51
Sick	8	0	49.15	49.28	98.43	95.23	85.26
Sick	8	1	157.30	59.95	217.25	97.35	94.52
Sick	8	2	171.16	55.56	226.72	97.61	86.53
Sick	8	3	194.73	50.95	245.68	98.67	93.20
Sick	9	0	23.72	72.17	95.88	96.29	85.83
Sick	9	1	108.30	57.22	165.52	96.82	94.24
Sick	9	2	102.30	62.92	165.22	97.88	84.64
Sick	9	3	153.59	55.92	209.50	98.14	94.95
Sick	9	4	139.73	54.40	194.12	98.41	97.12
Sick	9	5	147.59	54.33	201.92	98.41	97.12
Sick	10	0	23.72	55.95	79.67	96.55	92.07
Sick	10	1	110.15	51.92	162.07	97.35	84.36
Sick	10	2	163.30	54.69	217.99	97.08	94.38
Sick	10	3	179.02	54.47	233.49	98.14	94.95
Sick	10	4	179.02	55.79	234.81	97.61	94.66
Splice	1	0	106.07	345.38	451.44	74.29	78.15
Splice	1	1	707.12	178.06	885.18	90.28	92.28
Splice	1	2	830.43	191.91	1022.34	92.16	93.64
Splice	1	3	695.04	173.96	869.00	93.42	94.95
Splice	1	4	685.55	170.59	856.14	94.36	95.69
Splice	2	0	91.40	327.48	418.88	76.18	79.62
Splice	2	1	627.79	170.07	797.86	86.52	88.96
Splice	2	2	778.69	179.03	957.72	89.97	92.07
Splice	2	3	845.09	174.80	1019.89	92.79	94.41
Splice	2	4	830.43	170.90	1001.32	93.42	94.82
Splice	3	0	98.31	338.90	437.22	68.34	74.10
Splice	3	1	648.47	216.34	864.81	86.21	88.64
Splice	3	2	739.87	222.78	962.65	88.09	90.84
Splice	3	3	754.53	210.38	964.91	92.79	94.53
Splice	3	4	754.53	210.38	964.91	92.79	94.53
Splice	4	0	118.14	412.77	530.92	62.07	68.90
Splice	4	1	618.30	203.79	822.09	88.09	89.75
Splice	4	2	677.79	178.91	856.70	89.34	90.70
Splice	4	3	761.44	179.71	941.15	93.73	94.91
Splice	4	4	702.79	174.27	877.06	94.98	95.98
Splice	5	0	74.16	350.69	424.85	65.52	71.42
Splice	5	1	613.13	195.97	809.10	89.97	91.96
Splice	5	2	709.70	210.11	919.81	89.66	91.39
Splice	5	3	771.78	196.78	968.57	93.73	95.15
Splice	5	4	685.55	198.29	883.84	94.04	95.60
Splice	6	0	88.82	329.51	418.33	70.53	75.50
Splice	6	1	628.64	212.49	841.13	86.21	88.90
Splice	6	2	786.44	183.16	969.61	91.54	93.71
Splice	6	3	793.35	172.38	965.73	94.04	95.48
Splice	6	4	769.20	177.29	946.48	93.10	94.98
Splice	6	5	751.95	173.70	925.65	94.04	95.60
Splice	6	6	751.95	175.29	927.24	94.67	96.02

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Splice	7	0	106.07	326.59	432.66	66.77	72.79
Splice	7	1	586.39	194.18	780.57	86.21	88.78
Splice	7	2	786.44	180.91	967.35	92.48	94.10
Splice	7	3	709.70	189.42	899.12	92.48	94.69
Splice	7	4	749.36	183.47	932.83	94.36	95.93
Splice	8	0	88.82	303.99	392.81	76.18	80.28
Splice	8	1	663.13	192.43	855.57	89.03	91.71
Splice	8	2	665.72	196.75	862.47	88.40	91.41
Splice	8	3	818.35	194.91	1013.26	90.60	92.86
Splice	9	0	86.23	389.97	476.21	68.65	75.68
Splice	9	1	726.95	198.26	925.21	87.77	90.05
Splice	9	2	731.27	184.59	915.86	88.40	91.41
Splice	9	3	795.94	181.38	977.32	92.79	94.66
Splice	9	4	764.03	169.58	933.61	94.67	96.02
Splice	9	5	781.27	171.58	952.85	94.04	95.60
Splice	9	6	766.61	170.00	936.61	94.36	95.81
Splice	10	0	125.90	227.12	353.02	76.18	79.22
Splice	10	1	613.13	207.80	820.93	84.95	87.49
Splice	10	2	781.27	178.05	959.32	95.61	96.75
Splice	10	3	688.13	188.61	876.74	93.10	94.39
Splice	10	4	688.13	174.94	863.07	95.61	96.64
Splice	10	5	702.79	170.24	873.04	96.87	97.70
Splice	10	6	702.79	174.94	877.73	95.92	96.96
Waveform-5000	1	0	73.84	769.55	843.39	68.00	76.00
Waveform-5000	1	1	1124.85	469.92	1594.78	72.60	79.45
Waveform-5000	1	2	1819.59	413.67	2233.26	73.80	80.32
Waveform-5000	1	3	2113.52	426.93	2540.44	71.60	78.67
Waveform-5000	1	4	2291.66	411.43	2703.09	77.80	83.33
Waveform-5000	1	5	2371.82	391.00	2762.81	75.00	81.22
Waveform-5000	1	6	2327.28	391.27	2718.55	71.00	78.23
Waveform-5000	1	7	2541.05	403.51	2944.56	72.60	79.45
Waveform-5000	1	8	2532.14	396.98	2929.12	75.80	81.85
Waveform-5000	1	9	2630.12	392.18	3022.30	77.80	83.33
Waveform-5000	1	10	2674.65	390.15	3064.80	78.00	83.50
Waveform-5000	1	11	2407.45	380.81	2788.25	74.80	81.08
Waveform-5000	1	12	2416.35	378.62	2794.97	76.00	81.99
Waveform-5000	1	13	2514.33	395.62	2909.95	76.20	82.14
Waveform-5000	1	14	2505.42	398.49	2903.91	73.80	80.34
Waveform-5000	2	0	73.84	843.44	917.28	66.00	74.51
Waveform-5000	2	1	1302.99	436.46	1739.45	69.80	77.33
Waveform-5000	2	2	1596.92	452.83	2049.75	68.60	76.44
Waveform-5000	2	3	2113.52	427.17	2540.68	72.00	78.99
Waveform-5000	2	4	2354.00	432.05	2786.05	72.00	78.97
Waveform-5000	2	5	2247.12	411.35	2658.47	75.40	81.54
Waveform-5000	2	6	2478.70	408.57	2887.27	75.80	81.83
Waveform-5000	2	7	2220.40	424.01	2644.41	74.00	80.50
Waveform-5000	2	8	2273.84	403.69	2677.53	74.40	80.80
Waveform-5000	2	9	2603.40	404.48	3007.88	76.00	81.99
Waveform-5000	2	10	2398.54	407.87	2806.41	74.40	80.80
Waveform-5000	2	11	2514.33	416.63	2930.96	73.80	80.34
Waveform-5000	3	0	82.75	702.76	785.50	71.00	78.26

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D|H)	MDL	Test Acc	Test Auc
Waveform-5000	3	1	1311.90	458.48	1770.38	70.00	77.49
Waveform-5000	3	2	1988.82	433.03	2421.85	73.00	79.71
Waveform-5000	3	3	1953.19	427.10	2380.29	72.60	79.42
Waveform-5000	3	4	2113.52	437.10	2550.61	72.80	79.58
Waveform-5000	3	5	2282.75	401.49	2684.23	75.60	81.69
Waveform-5000	3	6	2345.10	392.39	2737.49	73.80	80.34
Waveform-5000	3	7	2505.42	419.07	2924.50	72.00	78.98
Waveform-5000	3	8	2523.24	416.07	2939.31	75.80	81.85
Waveform-5000	3	9	2380.72	411.38	2792.10	75.00	81.24
Waveform-5000	3	10	2460.89	437.33	2898.22	77.20	82.89
Waveform-5000	3	11	2737.00	384.74	3121.74	78.80	84.09
Waveform-5000	3	12	2487.61	413.40	2901.00	75.80	81.85
Waveform-5000	3	13	2487.61	436.14	2923.75	75.80	81.84
Waveform-5000	3	14	2523.24	441.05	2964.29	75.20	81.39
Waveform-5000	4	0	100.56	808.00	908.56	67.00	75.26
Waveform-5000	4	1	1249.55	450.60	1700.15	70.40	77.77
Waveform-5000	4	2	1650.36	426.05	2076.41	73.80	80.33
Waveform-5000	4	3	2220.40	444.17	2664.57	73.80	80.33
Waveform-5000	4	4	2264.94	404.77	2669.71	73.20	79.89
Waveform-5000	4	5	2264.94	409.29	2674.22	74.40	80.79
Waveform-5000	4	6	2371.82	413.45	2785.26	72.80	79.58
Waveform-5000	4	7	2291.66	407.09	2698.75	75.40	81.52
Waveform-5000	4	8	2362.91	412.40	2775.31	75.00	81.24
Waveform-5000	4	9	2558.86	414.84	2973.70	71.60	78.69
Waveform-5000	4	10	2532.14	414.61	2946.75	77.00	82.75
Waveform-5000	4	11	2549.96	408.80	2958.75	74.00	80.48
Waveform-5000	4	12	2398.54	414.55	2813.09	76.00	81.99
Waveform-5000	4	13	2505.42	418.66	2924.08	76.00	81.98
Waveform-5000	4	14	2487.61	421.72	2909.33	76.00	81.99
Waveform-5000	5	0	109.47	744.83	854.30	63.80	72.80
Waveform-5000	5	1	1231.74	483.23	1714.97	67.80	75.81
Waveform-5000	5	2	1881.94	415.72	2297.66	74.20	80.64
Waveform-5000	5	3	1997.73	447.41	2445.14	69.60	77.21
Waveform-5000	5	4	2362.91	432.98	2795.89	73.20	79.91
Waveform-5000	5	5	2238.21	430.19	2668.41	69.80	77.35
Waveform-5000	5	6	2345.10	423.57	2768.67	74.00	80.48
Waveform-5000	5	7	2389.63	436.80	2826.43	73.00	79.74
Waveform-5000	5	8	2523.24	436.23	2959.46	72.00	79.00
Waveform-5000	5	9	2523.24	411.83	2935.07	73.80	80.37
Waveform-5000	6	0	91.65	649.76	741.41	69.80	77.33
Waveform-5000	6	1	1302.99	461.54	1764.53	67.40	75.52
Waveform-5000	6	2	1971.01	413.22	2384.23	70.80	78.08
Waveform-5000	6	3	2149.15	425.95	2575.09	73.80	80.34
Waveform-5000	6	4	2460.89	461.11	2922.00	72.20	79.13
Waveform-5000	6	5	2469.79	425.88	2895.67	72.80	79.61
Waveform-5000	6	6	2674.65	436.40	3111.05	74.80	81.10
Waveform-5000	7	0	91.65	809.77	901.42	62.40	71.79
Waveform-5000	7	1	1285.18	426.16	1711.34	72.00	78.98
Waveform-5000	7	2	1855.22	418.92	2274.14	73.40	80.04
Waveform-5000	7	3	1971.01	428.02	2399.03	71.00	78.23
Waveform-5000	7	4	2256.03	398.53	2654.56	74.00	80.47

Continued on next page

Dataset	CV fold	Iteration	L(H)	L(D | H)	MDL	Test Acc	Test Auc
Waveform-5000	7	5	2487.61	408.43	2896.04	76.00	81.98
Waveform-5000	7	6	2327.28	400.33	2727.62	75.60	81.69
Waveform-5000	7	7	2576.68	398.21	2974.89	77.60	83.20
Waveform-5000	7	8	2576.68	430.46	3007.14	75.20	81.39
Waveform-5000	7	9	2630.12	404.49	3034.61	74.00	80.49
Waveform-5000	8	0	73.84	806.93	880.77	67.00	75.27
Waveform-5000	8	1	1249.55	485.50	1735.05	68.00	75.96
Waveform-5000	8	2	1694.89	436.06	2130.95	70.80	78.10
Waveform-5000	8	3	2166.96	407.08	2574.04	74.00	80.50
Waveform-5000	8	4	2327.28	415.97	2743.26	74.40	80.82
Waveform-5000	8	5	2256.03	436.52	2692.55	73.40	80.05
Waveform-5000	8	6	2478.70	425.14	2903.84	73.60	80.20
Waveform-5000	8	7	2362.91	427.19	2790.10	72.80	79.60
Waveform-5000	9	0	100.56	673.64	774.20	66.00	74.47
Waveform-5000	9	1	1035.78	446.54	1482.32	67.20	75.41
Waveform-5000	9	2	1783.96	396.12	2180.09	72.80	79.59
Waveform-5000	9	3	2193.68	422.02	2615.70	70.60	77.95
Waveform-5000	9	4	2247.12	394.68	2641.80	73.00	79.74
Waveform-5000	9	5	2371.82	412.83	2784.65	76.60	82.45
Waveform-5000	9	6	2354.00	413.00	2767.00	76.80	82.59
Waveform-5000	10	0	82.75	733.60	816.35	65.00	73.70
Waveform-5000	10	1	1347.53	471.82	1819.35	66.20	74.64
Waveform-5000	10	2	1846.31	423.47	2269.78	71.00	78.23
Waveform-5000	10	3	2077.89	415.53	2493.43	76.20	82.14
Waveform-5000	10	4	2264.94	387.61	2652.55	75.00	81.23
Waveform-5000	10	5	2175.87	390.16	2566.03	76.80	82.59

Accepted Papers D

The results of this dissertation were presented in two papers:

1. "Towards Windowing as a sub-sampling method for Distributed Data Mining" published in Research in Computing Science Journal.
2. "Windowing as a sub-sampling method for Distributed Data Mining" published in Mathematical and Computational Applications Journal.

Towards Windowing as a sub-sampling method for Distributed Data Mining.

David Martínez-Galicia[1], Alejandro Guerra-Hernández[1], Nicandro Cruz-Ramírez[1], Xavier Limón[2], and Francisco Grimaldo[3]

[1] Universidad Veracruzana, Centro de Investigación en Inteligencia Artificial, Sebastián Camacho No 5, Xalapa, Ver., México 91000.
davidgalicia@outlook.es, {aguerra,ncruz}@uv.mx
[2] Universidad Veracruzana, Facultad de Estadística e Informática, Av. Xalapa s/n, Xalapa, Ver., México 91000
hlimon@uv.mx
[3] Universitat de València, Departament d'Informàtica, Avinguda de la Universitat, s/n, Burjassot-València, España 46100
francisco.grimaldo@uv.es

Abstract. Windowing is a sub-sampling method that enables the induction of decision trees with large datasets. Using a small sample of the available training examples, the method can achieve levels of accuracy comparable or better than those obtained using the full available dataset. More relevant is the fact that Windowing-based strategies for Distributed Data Mining (DDM) have shown a correlation between the accuracy of the learned decision tree and the number of examples used to learn it, i.e., the higher the accuracy, the fewer examples used to induce the model. This paper corroborates that this behavior is also observed when adopting inductive algorithms of a different nature than C4.5 or ID3, the algorithms usually adopted when windowing, contributing to the use of Windowing as a general sub-sampling method for DDM. The paper also contributes exploring some metrics to the validation of the obtained sub-samples of examples.

Keywords: Sub-sampling · Windowing · Distributed Data Mining

1 Introduction

Windowing is a sub-sampling method that enabled the decision tree inductive algorithms ID3 [9–11] and C4.5 [12, 13] to cope with large datasets, i.e., those whose size precludes loading them in memory. Algorithm 1 defines the method: First, a window is created by extracting a small random sample of the available examples in the full dataset. The main step consists of inducing a model with the window and testing it on the remaining examples, such that all misclassified examples are moved to the window. This step iterates until a stop condition is reached, e.g., all the available examples are correctly classified or a desired level of accuracy is reached.

Algorithm 1 Windowing.

```
function WINDOWING(Examples)
    Window ← sample(Examples)
    Examples ← Examples − Window
    repeat
        stopCond ← true
        model ← induce(Window)
        for example ∈ Examples do
            if classify(model, example) ≠ class(example) then
                Window ← Window ∪ {example}
                Examples ← Examples − {example}
                stopCond ← false
    until stopCond
    return model
```

It has been argued [3] that the method offers three advantages: It copes well with memory limitations, reducing considerably the number of examples required to induce a model of acceptable accuracy. It offers an efficiency gain by reducing the time of convergence, specially when using a separate-and-conquer inductive algorithm, as FOIL [8], instead of the divide-and-conquer algorithms such as ID3 and C4.5. It offers an accuracy gain, specially in noiseless datasets, possibly explained by the fact that learning from a subset of examples may often result in a less over-fitting theory.

Although the lack of memory does not use to be an issue nowadays, similar concerns arise when mining big and/or distributed data. Windowing has been used as the core of a set of strategies for Distributed Data Mining (DDM) [6], obtaining consistent results with respect to the achievable accuracy and the number of examples required by the method. On the contrary, efficiency suffered for large datasets as the cost of testing the models in the remaining examples is not negligible. However, this is alleviated by using GPUs [5]. More relevant for this paper is the fact that the Windowing-based strategies shows a strong correlation (-0.8175845) between the accuracy of the learned decision trees and the number of examples used to induce them, i.e., the higher the accuracy obtained, the fewer the number of examples used to induce the model. Reductions are as big as the 90% of the available training data.

The objective of this work is to corroborate if such a correlation is observed when using inductive algorithms of different nature, so that the advantages of windowing as a sub-sampling method could be generalized beyond decision trees. For this, the paper is organized as follows: Section 2 introduces the adopted methodology; Section 3 presents the obtained results; and Section 4 discusses conclusions and futurework. A preliminary contribution of the paper is the study of some metrics to try to validate the obtained windows and to understand the way such sub-sampling works so efficiently in some cases.

2 Methodology

Because of our interest in distributed settings, JaCa-DDM [4] was adopted to run experiments. This tool [6] defines a set of Windowing-based strategies using J48, the Weka [14] implementation of C4.5, as inductive algorithm. Among them, Counter is the most similar to the original formulation of Windowing, excepting that: i) the dataset can be distributed in different sites, and ii) an auto-adjustable stop criteria with a established maximum number of iterations (10) is adopted. The parameters of the strategy, e.g., the maximum number of rounds, are adopted from the literature. The same configuration is used for all the experiments. The Counter strategy is tested on the datasets shown in Table 1, selected from the UCI [2] and MOA [1] repositories. They vary in the number of instances, attributes, and class' values; as well as in the type of the attributes. Some of them are affected by missing values.

Dataset	**Instances**	**Attribs**	**Types**	**Missing**	**Class**
Adult	48842	15	Mixed	Yes	2
Australian	690	15	Mixed	No	2
Breast	683	10	Numeric	No	2
Credit-g	1000	21	Mixed	No	2
Diabetes	768	9	Mixed	No	2
Ecoli	336	8	Numeric	No	8
German	1000	21	Mixed	No	2
Hypothyroid	3772	30	Mixed	Yes	4
Kr-vs-kp	3196	37	Numeric	No	2
Letter	20000	17	Mixed	No	26
Mushroom	8124	23	Nominal	Yes	2
Poker-lsn	829201	11	Mixed	No	10
Segment	2310	20	Numeric	No	7
Sick	3772	30	Mixed	Yes	2
Splice	3190	61	Nominal	No	3
Waveform5000	5000	41	Numeric	No	3

Table 1. Datasets, adopted from UCI and MOA.

Apart from J48, the Counter strategy will be tested using the Weka implementations of Naive Bayes, jRip, Multi-Perceptron, and SMO as inductive algorithms. A 10-fold stratified cross-validation is run on each dataset, observing the average accuracy of the obtained models and the average percentage of original dataset used to induce the model, i.e., 100% means the full original dataset was used. All experiments were executed on a Intel Core i5-8300H at 2.3GHz, up to 3.9GHz with 8Gb DDR4. 8 distributed sites were simulated on this machine.

[4] https://github.com/xl666/jaca-ddm

In order to understand the performed sub-sampling, the following measures were used to compare the obtained window and the original dataset:

- The Kullback-Leibler divergence (D_{KL}) [4] is defined as:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) log_2(\frac{P(x)}{Q(x)})$$

  where $P(x)$ is the full dataset class distribution and $Q(x)$ the window class distribution. Instead of using a model to represent a conditional distribution of variables, as usual, we focus on the class distribution, computed as the marginal probability. Values closer to zero reflect higher similarity.
- Sim_1 [15] is a similarity measure between datasets defined as:

$$sim_1(D_i, D_j) = \frac{|Item(D_i) \cap Item(D_j)|}{|Item(D_i) \cup Item(D_j)|}$$

  where D_i is the window and D_j is the full dataset; and $Item(D)$ denotes the set of pairs attribute-value occurring in D. Values closer to one reflect higher similarity.
- *Red* [7] measures redundancy in a dataset in terms of conditional population entropy (CPE), defined as:

$$CPE = -\sum_{i=1}^{n_c} p(c_i) \sum_{a=1}^{n_a} \sum_{v=1}^{n_{v_a}} p(x_{a,v}|c_i) log_2 p(x_{a,v}|c_i)$$

  where n_c is the number of classes, n_a is the number of attributes, and n_{v_a} is the number of values for the attribute a. c_i stands for the $i-th$ class and $x_{a,v}$ represents the $v-th$ value of attribute a. CPE can be normalized [3] in such a way that values closer to zero reflect lower redundancy:

$$Red = 1 - \frac{CPE}{\sum_{a=1}^{n_a} log_2 n_{v_a}}$$

3 Results

Figure 1 shows a strong negative correlation between the percentage of training instances used to induce the models and their accuracy, independently of the adopted inductive algorithm. This reproduces the results for J48 reported in literature [6] and corroborates that under Windowing, in general, the models with higher accuracy require less examples to be induced. However, accuracy is affected by the adopted inductive algorithm, e.g., Poker-lsn is approached very well by J48 (99.75 $\pm$ 0.07 of accuracy) requiring few examples (5% of the full dataset); while Naive Bayes is not quite successful in this case (60.02 $\pm$ 0.42 of accuracy) requiring more examples (59%). This behavior is also observed between jRip and MultiPerceptron for Hypothyroid; and between SMO and jRip for Waveform5000.

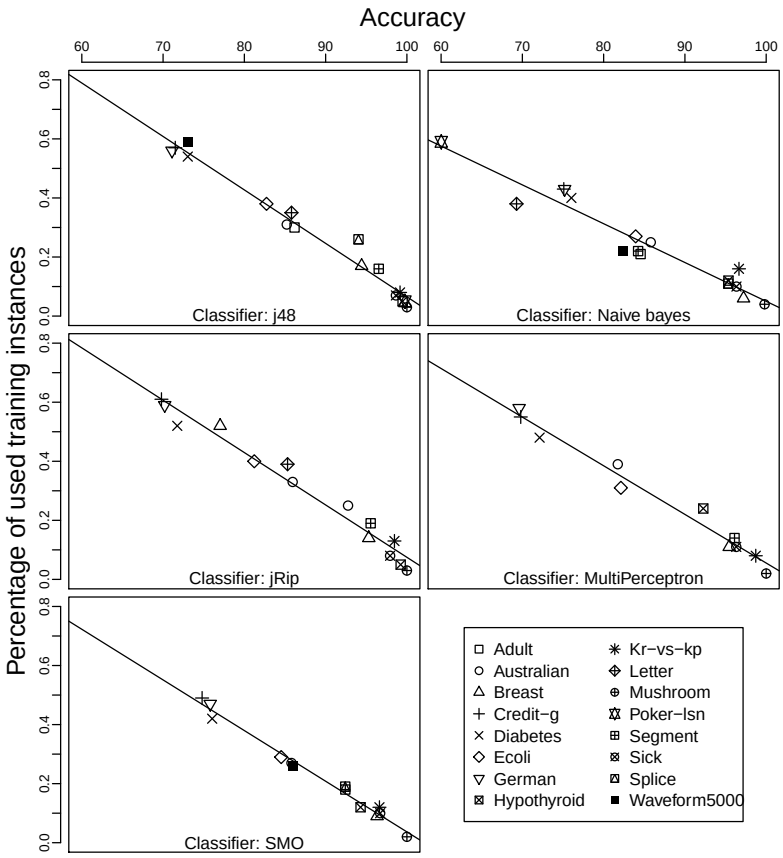


Fig. 1. Correlation between accuracy and percentage of used training examples. J48 = -0.98, NB = -0.96, jRip = -0.98, MP = -0.98 and SMO = -0.99.

Table 2 shows the accuracy results in detail while Table 3 show the number of used examples results, in terms of the percentage of the full dataset used for each inductive algorithm. Although not shown because of the available space, accuracies are comparable to those obtained without using Windowing, i.e., using the 100% of the available data to induce the models. Big datasets, as Adult, Letter, Poker-Isn, Splice, and Waveform5000 did not finish on reasonable time when using jRip, MultiPerceptron and SMO, with and without Windowing. In such cases, results are reported as not available (na). This might be solved by running the experiments in a real cluster of 8 nodes, instead of simulating the sites in a single machine, as done here, but it is not relevant for the purposes of this work.

	J48	NB	jRip	MP	SMO
Adult	86.17 ± 0.55	84.54 ± 0.62	na	na	na
Australian	85.21 ± 4.77	85.79 ± 4.25	85.94 ± 3.93	81.74 ± 6.31	85.80 ± 4.77
Breast	94.42 ± 3.97	97.21 ± 2.34	95.31 ± 2.75	95.45 ± 3.14	96.33 ± 3.12
Credit-g	71.50 ± 5.81	75.10 ± 2.60	69.80 ± 3.71	69.80 ± 5.63	74.80 ± 5.98
Diabetes	73.03 ± 3.99	76.03 ± 4.33	71.74 ± 7.67	72.12 ± 4.00	76.04 ± 3.51
Ecoli	82.72 ± 6.81	83.93 ± 7.00	81.22 ± 6.63	82.12 ± 7.49	84.53 ± 4.11
German	71.10 ± 5.40	75.20 ± 2.82	70.20 ± 3.85	69.60 ± 4.84	75.80 ± 3.12
Hypothyroid	99.46 ± 0.17	95.36 ± 0.99	99.23 ± 0.48	92.26 ± 2.75	94.30 ± 0.53
Kr-vs-kp	99.15 ± 0.66	96.65 ± 0.84	98.46 ± 0.95	98.72 ± 0.54	96.62 ± 0.75
Letter	85.79 ± 1.24	69.28 ± 1.26	85.31 ± 1.06	na	na
Mushroom	100.00 ± 0.00	99.80 ± 0.16	100.00 ± 0.00	100.00 ± 0.00	100.0 ± 0.00
Poker-lsn	99.75 ± 0.07	60.02 ± 0.42	na	na	na
Segment	96.53 ± 1.47	84.24 ± 1.91	95.54 ± 1.55	96.10 ± 1.15	92.42 ± 1.87
Sick	98.64 ± 0.53	96.34 ± 1.44	97.93 ± 0.95	96.32 ± 1.04	96.71 ± 0.77
Splice	94.04 ± 0.79	95.32 ± 1.07	92.75 ± 2.11	na	92.41 ± 1.34
Waveform5000	73.06 ± 2.55	82.36 ± 1.64	77.02 ± 1.59	na	85.94 ± 1.32

Table 2. Accuracies obtained from 10-fold cross validation (na = not available).

	J48	NB	jRip	MP	SMO
Adult	0.30 ± 0.01	0.21 ± 0.00	na	na	na
Australian	0.31 ± 0.02	0.25 ± 0.01	0.33 ± 0.02	0.39 ± 0.04	0.27 ± 0.01
Breast	0.17 ± 0.01	0.06 ± 0.00	0.14 ± 0.01	0.11 ± 0.01	0.09 ± 0.01
Credit-g	0.57 ± 0.03	0.43 ± 0.01	0.61 ± 0.01	0.55 ± 0.04	0.49 ± 0.01
Diabetes	0.54 ± 0.05	0.40 ± 0.02	0.52 ± 0.04	0.48 ± 0.03	0.42 ± 0.02
Ecoli	0.38 ± 0.03	0.27 ± 0.01	0.40 ± 0.03	0.31 ± 0.03	0.29 ± 0.02
German	0.56 ± 0.04	0.43 ± 0.01	0.59 ± 0.02	0.58 ± 0.02	0.47 ± 0.02
Hypothyroid	0.05 ± 0.00	0.12 ± 0.01	0.05 ± 0.00	0.24 ± 0.01	0.12 ± 0.01
Kr-vs-kp	0.08 ± 0.01	0.16 ± 0.01	0.13 ± 0.00	0.08 ± 0.00	0.12 ± 0.00
Letter	0.35 ± 0.02	0.38 ± 0.00	0.39 ± 0.01	na	na
Mushroom	0.03 ± 0.00	0.04 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
Poker-lsn	0.05 ± 0.00	0.59 ± 0.00	na	na	na
Segment	0.16 ± 0.01	0.22 ± 0.01	0.19 ± 0.01	0.14 ± 0.01	0.18 ± 0.00
Sick	0.07 ± 0.00	0.10 ± 0.01	0.08 ± 0.00	0.11 ± 0.01	0.10 ± 0.00
Splice	0.26 ± 0.01	0.11 ± 0.00	0.25 ± 0.01	na	0.19 ± 0.00
Waveform5000	0.59 ± 0.02	0.22 ± 0.01	0.52 ± 0.00	na	0.26 ± 0.01

Table 3. Percentage of the full dataset used for induction (na = not available).

The Kullback-Leibler divergence coefficient between the windows and the full datasets was close to zero in all cases ($D_{KL} < 0.25$), evidencing that the class distribution of the windows is very similar to that observed in the full datasets. However it does not seem to be a correlation between this coefficient and the obtained accuracy, e.g., Mushroom has zero as divergence coefficient and 100%

of accuracy, but Waveform5000 has similar divergence but considerable lower accuracy.

Table 4 shows the results for sim_1, suggesting that the windows for Australian, Breast, German, Letter, Kr-vs-Kp, and Poker-lsn conserve all the values for their attributes observed in the full datasets; while Adult and Segment have problems achieving this. As in the previous case, this notion of similarity neither seems to correlate with the observed accuracy, e.g., Segment.

	j48	**NB**	**jRip**	**MP**	**SMO**
Adult	0.39±0.01	0.29±0.00	na	na	na
Australian	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Breast	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Credit-g	0.63±0.03	0.51±0.01	0.69 ± 0.01	0.63 ± 0.04	0.58 ± 0.01
Diabetes	0.73±0.04	0.63±0.02	0.72 ± 0.03	0.69 ± 0.02	0.64 ± 0.01
Ecoli	0.77±0.03	0.65±0.02	0.78 ± 0.02	0.69 ± 0.04	0.65 ± 0.03
German	1.00±0.00	1.00±0.00	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00
Hypothyroid	0.45±0.01	1.00±0.01	0.48 ± 0.01	0.68 ± 0.01	0.59 ± 0.01
Kr-vs-kp	1.00±0.01	0.97±0.01	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
Letter	0.99±0.01	0.99±0.01	0.98 ± 0.00	na	na
Mushroom	0.97±0.02	0.99±0.01	0.98 ± 0.00	0.97 ± 0.01	0.97 ± 0.01
Poker-lsn	1.00±0.00	1.00±0.00	na	na	na
Segment	0.28±0.01	0.32±0.01	0.31 ± 0.01	0.25 ± 0.01	0.28 ± 0.00
Sick	0.57±0.02	0.58±0.01	0.59 ± 0.01	0.60 ± 0.02	0.60 ± 0.01
Splice	0.97±0.04	0.96±0.05	0.97 ± 0.03	na	0.96 ± 0.04
Waveform5000	0.93±0.01	0.71±0.01	0.90 ± 0.00	na	0.76 ± 0.01

Table 4. Table of similarity measure sim_1 using the 10-folds cross-validation windows.

Red shows consistently the same values for the windows and the full datasets, meaning that both of them have very similar levels of redundancy. Given the nature of Windowing this can be a little bit surprising, since the window is expected to be less redundant than the full dataset because it does not include examples already covered by the induced models. But *Red* measures the information value given the information about the class values, an intrinsic property of the data set; while the redundancy reduction expected by Windowing is a property of a dataset given a classifier. This behavior of *Red*, reported in literature [3], suggests that a different measure for redundancy should be adopted.

4 Conclusions and future work

The correlation between the accuracy of the models obtained by Windowing and the number of examples used for this task was corroborated, independently of the adopted inductive algorithm, i.e., high accurate models require fewer examples to be learned. The metrics suggest that the windows have a class distribution

very similar to the full datasets, as well as the same items (attribute-value pairs). They also have very similar intrinsic redundancy. Unfortunately, such similarities are not enough to explain the success of the technique since they do not correlate with the obtained accuracy of the models.

Up to our knowledge, this is the first comparative study of Windowing in this respect. Future work requires finding a metric reflecting the notion of redundancy in terms of the set of covered examples to quantify the efficiency of Windowing as a sub-sampling method. Also, observing the evolution of the windows through the whole process seems pertinent to enhance our understanding of Windowing.

References

1. Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: massive online analysis. *J. Mach. Learn. Res.*, 11:1601–1604, 2010.
2. Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
3. Johannes Fürnkranz. Integrative windowing. *Journal of Artificial Intelligence Research*, 8:129–164, 1998.
4. Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
5. Xavier Limón, Alejandro Guerra-Hernández, Nicandro Cruz-Ramírez, Héctor Gabriel Acosta-Mesa, and Francisco Grimaldo. A windowing strategy for distributed data mining optimized through GPUs. *Pattern Recognition Letters*, 93(Suplement C):23–30, July 2017.
6. Xavier Limón, Alejandro Guerra-Hernández, Nicandro Cruz-Ramírez, and Francisco Grimaldo. Modeling and implementing distributed data mining strategies in JaCa-DDM. *Knowledge and Information Systems*, 60(1):99–143, 2019.
7. Martin Møller. Supervised learning on large redundant training sets. *International Journal of Neural Systems*, 4(1):15–25, 1993.
8. J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
9. John Ross Quinlan. Induction over large data bases. Technical Report STAN-CS-79-739, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, CA, USA, May 1979.
10. John Ross Quinlan. Learning efficient classification procedures and their application to chess en games. In Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors, *Machine Learning*, volume I, chapter 15, pages 463 – 482. Morgan Kaufmann, San Francisco (CA), 1983.
11. John Ross Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
12. John Ross Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, San Mateo, CA., USA, 1993.
13. John Ross Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.
14. Ian H Witten, Eibe Frank, and Mark A Hall. *Data Mining: Practical Machine Learning Toools and Techniques*. Morgan Kaufmann Publishers, Burlington, MA., USA, 2011.
15. Shichao Zhang, Chengqi Zhang, and Xindong Wu. *Knowledge Discovery in Multiple Databases*. Advanced Information and Knowledge Processing. Springer-Verlag London, Limited, London, UK, 2004.

Mathematical and Computational Applications

MDPI

Article

Windowing as a Sub-Sampling Method for Distributed Data Mining

David Martínez-Galicia [1,*], Alejandro Guerra-Hernández [1], Nicandro Cruz-Ramírez [1], Xavier Limón [2] and Francisco Grimaldo [3]

[1] Centro de Investigación en Inteligencia Artificial, Universidad Veracruzana, Sebastián Camacho No 5, Xalapa, Veracruz, México 91000, Mexico; aguerra@uv.mx (A.G.-H.); ncruz@uv.mx (N.C.-R.)
[2] Facultad de Estadística e Informática, Universidad Veracruzana, Av. Xalapa s/n, Xalapa, Veracruz, México 91000, Mexico; hlimon@uv.mx
[3] Departament d'Informàtica, Universitat de València, Avinguda de la Universitat, s/n, Burjassot-València, 46100 València, Spain; francisco.grimaldo@uv.es
* Correspondence: davidgalicia@outlook.es

Received: 31 May 2020; Accepted: 29 June 2020; Published: 30 June 2020

Abstract: Windowing is a sub-sampling method, originally proposed to cope with large datasets when inducing decision trees with the ID3 and C4.5 algorithms. The method exhibits a strong negative correlation between the accuracy of the learned models and the number of examples used to induce them, i.e., the higher the accuracy of the obtained model, the fewer examples used to induce it. This paper contributes to a better understanding of this behavior in order to promote windowing as a sub-sampling method for Distributed Data Mining. For this, the generalization of the behavior of windowing beyond decision trees is established, by corroborating the observed negative correlation when adopting inductive algorithms of different nature. Then, focusing on decision trees, the windows (samples) and the obtained models are analyzed in terms of Minimum Description Length (MDL), Area Under the ROC Curve (AUC), Kulllback–Leibler divergence, and the similitude metric Sim1; and compared to those obtained when using traditional methods: random, balanced, and stratified samplings. It is shown that the aggressive sampling performed by windowing, up to 3% of the original dataset, induces models that are significantly more accurate than those obtained from the traditional sampling methods, among which only the balanced sampling is comparable in terms of AUC. Although the considered informational properties did not correlate with the obtained accuracy, they provide clues about the behavior of windowing and suggest further experiments to enhance such understanding and the performance of the method, i.e., studying the evolution of the windows over time.

Keywords: sub-sampling; windowing; distributed data mining

1. Introduction

Windowing is a sub-sampling method that enabled the decision tree inductive algorithms ID3 [1–3] and C4.5 [4,5] to cope with large datasets, i.e., those whose size precludes loading them in memory. Algorithm 1 defines the method: First, a window is created by extracting a small random sample of the available examples in the full dataset. The main step consists of inducing a model with that window and of testing it on the remaining examples, such that all misclassified examples are moved to the window. This step iterates until a stop condition is reached, e.g., all the available examples are correctly classified or a desired level of accuracy is reached.

Algorithm 1 Windowing.

Require: *Examples* {The original training set}
Ensure: *Model* {The induced model}
1: $Window \leftarrow sample(Examples)$
2: $Examples \leftarrow Examples - Window$
3: **repeat**
4: $stopCond \leftarrow true$
5: $model \leftarrow induce(Window)$
6: **for** $example \in Examples$ **do**
7: **if** $classify(model, example) \neq class(example)$ **then**
8: $Window \leftarrow Window \cup \{example\}$
9: $Examples \leftarrow Examples - \{example\}$
10: $stopCond \leftarrow false$
11: **end if**
12: **end for**
13: **until** *stopCond*
14: **return** *model*

Despite Wirth and Catlett [6] publishing an early critic about the computational cost of windowing and its inability to deal with noisy domains, Fürnkranz [7] argues that this method still offers three advantages: a) it copes well with memory limitations, reducing considerably the number of examples required to induce a model of acceptable accuracy; b) it offers an efficiency gain by reducing the time of convergence, specially when using a separate-and-conquer inductive algorithm, as FOIL [8], instead of the divide-and-conquer algorithms such as ID3 and C4.5., and; c) it offers an accuracy gain, specially in noiseless datasets, possibly explained by the fact that learning from a subset of examples may often result in a less over-fitting theory.

Even when the lack of memory is not usually an issue nowadays, similar concerns arise when mining big and/or distributed data, i.e., the impossibility or inconvenience of using all the available examples to induce models. Windowing has been used as the core of a set of strategies for Distributed Data Mining (DDM) [9] obtaining good accuracy results, consistent with the expected achievable accuracy and number of examples required by the method. On the contrary, efficiency suffers for large datasets as the cost of testing the models in the remaining examples is not negligible (i.e., the for loop in Algorithm 1, line 6), although it can be alleviated by using GPUs [10]. More relevant for this paper is the fact that these Windowing-based strategies based on J48, the Weka [11] implementation of C4.5, show a strong correlation (-0.8175845) between the accuracy of the learned decision trees and the number of examples used to induce them, i.e., the higher the accuracy obtained, the fewer the number of examples used to induce the model. The windows in this method can be seen as samples and reducing the size of the training sets, even up to a 95% of the available training data, still enables accuracy values above 95%.

These promising results encourage the adoption of windowing as a sub-sampling method for Distributed Data Mining. However, they suggest some issues that must be solved for such adoption. The first one is the generalization of windowing beyond decision trees. Does windowing behave similarly when using different models and inductive algorithms? The first contribution of this paper is to corroborate the correlation between accuracy and the size of the window, i.e., the number of examples used to induce the model, when using inductive algorithms of different nature, showing that the advantages of windowing as a sub-sampling method can be generalized beyond decision trees. The second issue is the need of a deeper understanding of the behavior of windowing. How is that such a big reduction in the number of training examples, maintains acceptable levels of accuracy? This is particularly interesting as we have pointed out that high levels of accuracy correlate with smaller windows. The second contribution of the paper is thus to approach such a question in terms of the informational properties of both the windows and the models obtained by the method. These properties do not unfortunately correlate with the obtained accuracy of windowing and suggest the

study of the evolution of the windows over as future work. Finally, a comparison with traditional methods as random, stratified, and balanced samplings, provides a better understanding of windowing and evaluates its adoption as an alternative sampling method. Under equal conditions, i.e., same original full dataset and size of the sample, windowing shows to be significantly more accurate than the traditional samplings and comparable to balanced sampling in terms of AUC. The paper is organized as follows: Section 2 introduces the adopted materials and methods; Section 3 presents the obtained results; and Section 4 discusses conclusions and future work.

2. Materials and Methods

This section describes the implementation of windowing used in this work, as included in JaCa-DDM; the datasets used in experimentation; and the experiments themselves.

2.1. Windowing in JaCa-DDM

Because of our interest in Distributed Data Mining settings, JaCa-DDM (https://github.com/xl666/jaca-ddm) was adopted to run our experiments. This tool [9] defines a set of windowing-based strategies using J48, the Weka [11] implementation of C4.5, as inductive algorithm. Among them, the Counter strategy is the most similar to the original formulation of windowing, with the exception of:

1. The dataset may be distributed in different sites, instead of the traditional approach based on a single dataset in a single site.
2. The loop for collecting the misclassified examples to be added to the window is performed by a set of agents using copies of the model distributed among the available sites, in a round-robin fashion.
3. The initial window is a stratified sample, instead of a random one.
4. An auto-adjustable stop criteria is combined with a configurable maximum number of iterations.

The configuration of the strategy (Table 1) used for all the experiments reported in this paper, is adopted from the literature [10].

Table 1. Configuration of the counter strategy. Adopted from Limón *et al.* [10].

Parameter	Value
Classifier	J48
Pruning	True
Number of nodes	8
Maximum number of rounds	15
Initial percentage for the window	0.20
Validation percentage for the test	0.25
Change step of accuracy every round	0.35

2.2. Datsets

Table 2 lists the datasets selected from the UCI [12] and MOA [13] repositories to conduct our experiments. They vary in the number of instances, attributes, and class' values; as well as in the type of the attributes. Some of them are affected by missing values. The literature [10] reports experiments on larger datasets, up to 4.8×10^6 instances, exploiting GPUs. However, datasets with higher dimensions are problematic, e.g., imdb-D with 1002 attributes does not converge using the Counter strategy.

Table 2. Datasets, adopted from UCI and MOA.

Dataset	Instances	Attributes	Attribute type	Missing values	Classes
Adult	48842	15	Mixed	Yes	2
Australian	690	15	Mixed	No	2
Breast	683	10	Numeric	No	2
Diabetes	768	9	Mixed	No	2
Ecoli	336	8	Numeric	No	8
German	1000	21	Mixed	No	2
Hypothyroid	3772	30	Mixed	Yes	4
Kr-vs-kp	3196	37	Numeric	No	2
Letter	20000	17	Mixed	No	26
Mushroom	8124	23	Nominal	Yes	2
Poker-lsn	829201	11	Mixed	No	10
Segment	2310	20	Numeric	No	7
Sick	3772	30	Mixed	Yes	2
Splice	3190	61	Nominal	No	3
Waveform5000	5000	41	Numeric	No	3

2.3. Experiments

Two experiments were designed to cope with the issues approached by this work, i.e., the generalization of windowing beyond decision trees; a deeper understanding of its behavior in informational terms; and the comparison with traditional sampling methods. All of them were executed on a Intel Core i5-8300H at 2.3GHz, up to 3.9GHz with 8Gb DDR4. 8 distributed sites were simulated on this machine. JaCa-DDM also allows the adoption of real distributed sites over a network, but the aspects of windowing we study here, are not affected by simulating distribution.

2.3.1. On the Generalization of windowing

The first experiment seeks to corroborate the correlation between the accuracy of the learned model and the amount of instances used to induce the model. It attempts to provide practical evidence about the generalization of windowing. For this, different Weka classifiers are adopted that replace J48. JaCa-DDM allows easy replacement and configuration of the new classifier artifacts of the system, namely:

Naive Bayes. A probabilistic classifier based on Bayes' theorem with a strong assumption of independence among attributes [14].

jRip. An inductive rule learner based on RIPPER that builds a set of rules while minimizing the amount of error [15].

Multilayer-perceptron. A multi-layer perceptron trained by backpropagation with sigmoid nodes except for numeric classes, in which case the output nodes become unthresholded linear units [16].

SMO. An implementation of John Platt's sequential minimal optimization algorithm for training a support vector classifier [17].

All classifiers are induced by running a 10-fold stratified cross-validation on each dataset, then observing the average accuracy of the obtained models and the average percentage of the original dataset used to induce the model, i.e., 100% means the full original dataset was used to create the window.

2.3.2. On the Properties of Samples and Models Obtained by Windowing

The second experiment pursues a deeper understanding of the informational properties of the computed models, as well as those of the samples obtained by Windowing, i.e., the final windows. For this, given the positive results of the first experiment, we focus exclusively on decision trees (J48), for

which different metrics to evaluate performance, complexity and data compression are well known. They include:

- The model accuracy defined as the percentage of correctly classified instances.

$$\frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

where TP, TN, FP and FN respectively stand for the true positive, true negative, false positive, and false negative classifications using the test data.

- The metric AUC defined as the probability of a random instance to be correctly classified [18].

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{2}$$

Even though this measure was conceived for binary classification problems. Foster Provost [19] proposes an implementation for multi-class problems based in the weighted average of AUC metrics for every class using a one-against-all approach, and the weight for every AUC is calculated as the class' appearance frequency in the data $p(c_i)$.

$$AUC_{total} = \sum_{c_i \in C} AUC(c_i) \cdot p(c_i) \tag{3}$$

- The MDL principle states that the best model to infer from a dataset is the one which minimizes the sum of the length of the model $L(H)$, and the length of the data when encoded using the theory as a predictor for the data $L(D|H)$ [20].

$$MDL = L(H) + L(D|H) \tag{4}$$

For decision trees, Quinlan [21] proposes the next definition:

1. The number of bits needed to encode a tree is:

$$L(H) = n_{nodes} * (1 + ln(n_{attributes})) + n_{leaves}(1 + ln(n_{c1asses})) \tag{5}$$

where n_{nodes}, $n_{attributes}$, n_{leaves} and $n_{c1asses}$ stand for the number of nodes, attributes, leaves and classes. This encoding uses a recursive top-down, depth-first procedure, where a tree which is not a leaf is encoded by a sequence of 1, the attribute code at his root, and the respective encodings of the subtrees. If a tree or subtree is a leaf, its enconding is a sequence of 0, and the class code.

2. The number of bits needed to encode the data using the decision tree is:

$$L(D|H) = \sum_{l \in Leaves} log_2(b + 1) + log_2\left(\binom{n}{k}\right) \tag{6}$$

where n is the number of instances, k is the number of positives instances for binary classification and b is a known a priori upper bound on k, typically $b = n$. For non-binary classification, Quinlan proposes a iterative approach where exceptions are sorted by their frequency, and then codified with the previous formula.

- The Kullback–Leibler divergence (D_{KL}) [22] is defined as:

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) log_2 \left(\frac{P(x)}{Q(x)} \right) \quad (7)$$

where P and Q are probability distributions for the full dataset and the window, both are defined on the same probability space X, and x represents a class in the distribution. Instead of using a model to represent a conditional distribution of variables, as usual, we focus on the class distribution, computed as the marginal probability. Values closer to zero reflect higher similarity.

- Sim_1 [23] is a similarity measure between datasets defined as:

$$sim_1(D_i, D_j) = \frac{|Item(D_i) \cap Item(D_j)|}{|Item(D_i) \cup Item(D_j)|} \quad (8)$$

where D_i is the window and D_j is the full dataset; and $Item(D)$ denotes the set of pairs attribute-value occurring in D. Values closer to one reflect higher similarity.

These metrics are used to compare the sample (the window) and the model computed by windowing, against those obtained as follows, once a random sample of the original data set is reserved as test set:

- Without sampling, using all the available data to induce the model.
- By Random sampling, where any instance has the same selection probability [24].
- By Stratified random sampling, where the instances are subdivided by their class into subgroups, the number of selected instances per subgroup is defined as the division of the sample size by the number of instances [24].
- By Balanced random sampling, as stratified random sampling, the instances are subdivided by their class into subgroups, but the number of selected instances per subgroup is defined as the division of the sample size by the number of subgroups, this allows the same number of instances per class [24].

Ten repetitions of 10-fold stratified cross-validation are run on each dataset. For a fair comparison, all the samples have the size of the window being compared. Statistical validity of the results is established following the method proposed by Demšar [25]. This approach enables the comparison of multiple algorithms on multiple data sets. It is based on the use of the Friedman test with a corresponding post-hoc test. Let R_i^j be the rank of the j^{th} of k algorithms on the i^{th} of N data sets. The Friedman test [26,27] compares the average ranks of algorithms, $R_j = \frac{1}{N} \sum_i R_i^j$. Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks R_j should be equal, the Friedman statistic:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (9)$$

is distributed according to χ_F^2 with $k-1$ degrees of freedom, when N and k are big enough ($N > 10$ and $k > 5$). For a smaller number of algorithms and data sets, exact critical values have been computed [28]. Iman and Davenport [29] showed that Friedman's χ_F^2 is undesirably conservative and derived an adjusted statistic:

$$F_f = \frac{(N-1) \times \chi_F^2}{N \times (k-1) - \chi_F^2} \quad (10)$$

which is distributed according to the F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. If the null hypothesis of similar performances is rejected, then the Nemenyi post-hoc test is realized

for pairwise comparisons. The performance of two classifiers is significantly different if their corresponding average ranks differ by at least the critical difference:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{11}$$

where critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$.

For the comparison of multiple classifiers, the results of the post-hoc tests can be visually represented with a simple critical distance diagram. This type of visualization will be described in the Statistical Tests in Section 3.

3. Results

Results are organized accordingly to the following issues:

- Generalization of the behavior of windowing, i.e., high accuracy correlating with fewer training examples used to induce the model, when other inductive algorithms, apart of J48, are adopted.
- Informational properties of the samples obtained by different methods, based on the Kullback–Leibler divergence and the attribute-value similitude.
- Properties of the models induced with the samples, in terms of their size, complexity, and data compression, which supplies information about their data fitting capacity.
- Predictive performance of the induced models in terms of accuracy and the AUC.
- Statistical tests about significant gains produced by windowing using the former metrics.

3.1. Windowing Generalization

Figure 1 shows a strong negative correlation between the number of training instances used to induce the models, expressed as a percentage with respect to the totality of available examples, and the accuracy of the induced model. Such correlation exists, independently of the adopted inductive algorithm. These results are consistent with the behavior of windowing when using J48, as reported in the literature [9] and corroborates that under windowing, in general, the models with higher accuracy use less examples to be induced.

However, accuracy is affected by the adopted inductive algorithm, e.g., Hypothyroid is approached very well by jRip (99.23 ± 0.48 of accuracy) requiring few examples (5% of the full dataset); while Multilayer-Perceptron is not quite successful in this case (92.26 ± 2.75 of accuracy) requiring more examples (24%). This behavior is also observed between SMO and jRip for Waveform5000. These observations motivated analyzing the properties of the samples and induced models, as described in the following subsections. Table 3 shows the accuracy results in detail and Table 4 shows the number of examples used to induce the models, best results are highlighted in gray. Appendix A shows the accuracy values for models without using windowing under a 10-fold cross-validation. Windowing accuracies are comparable to those obtained without using windowing. Table 7 also corroborate this this for the J48 classifier.

Table 3. Average windowing accuracy under a 10-fold cross validation (na = not available).

	J48	NB	jRip	MP	SMO
Adult	86.17 ± 0.55	84.54 ± 0.62	na	na	na
Australian	85.21 ± 4.77	85.79 ± 4.25	85.94 ± 3.93	81.74 ± 6.31	85.80 ± 4.77
Breast	94.42 ± 3.97	97.21 ± 2.34	95.31 ± 2.75	95.45 ± 3.14	96.33 ± 3.12
Diabetes	73.03 ± 3.99	76.03 ± 4.33	71.74 ± 7.67	72.12 ± 4.00	76.04 ± 3.51
Ecoli	82.72 ± 6.81	83.93 ± 7.00	81.22 ± 6.63	82.12 ± 7.49	84.53 ± 4.11
German	71.10 ± 5.40	75.20 ± 2.82	70.20 ± 3.85	69.60 ± 4.84	75.80 ± 3.12
Hypothyroid	99.46 ± 0.17	95.36 ± 0.99	99.23 ± 0.48	92.26 ± 2.75	94.30 ± 0.53
Kr-vs-kp	99.15 ± 0.66	96.65 ± 0.84	98.46 ± 0.95	98.72 ± 0.54	96.62 ± 0.75
Letter	85.79 ± 1.24	69.28 ± 1.26	85.31 ± 1.06	na	na
Mushroom	100.00 ± 0.00	99.80 ± 0.16	100.00 ± 0.00	100.00 ± 0.00	100.0 ± 0.00
Poker-lsn	99.75 ± 0.07	60.02 ± 0.42	na	na	na
Segment	96.53 ± 1.47	84.24 ± 1.91	95.54 ± 1.55	96.10 ± 1.15	92.42 ± 1.87
Sick	98.64 ± 0.53	96.34 ± 1.44	97.93 ± 0.95	96.32 ± 1.04	96.71 ± 0.77
Splice	94.04 ± 0.79	95.32 ± 1.07	92.75 ± 2.11	na	92.41 ± 1.34
Waveform5000	73.06 ± 2.55	82.36 ± 1.64	77.02 ± 1.59	na	85.94 ± 1.32

Table 4. Average size of the final window (the sample) under a 10-fold cross validation, in terms of the percentage of the full dataset used for induction (na = not available).

	J48	NB	jRip	MP	SMO
Adult	0.30 ± 0.01	0.21 ± 0.00	na	na	na
Australian	0.31 ± 0.02	0.25 ± 0.01	0.33 ± 0.02	0.39 ± 0.04	0.27 ± 0.01
Breast	0.17 ± 0.01	0.06 ± 0.00	0.14 ± 0.01	0.11 ± 0.01	0.09 ± 0.01
Diabetes	0.54 ± 0.05	0.40 ± 0.02	0.52 ± 0.04	0.48 ± 0.03	0.42 ± 0.02
Ecoli	0.38 ± 0.03	0.27 ± 0.01	0.40 ± 0.03	0.31 ± 0.03	0.29 ± 0.02
German	0.56 ± 0.04	0.43 ± 0.01	0.59 ± 0.02	0.58 ± 0.02	0.47 ± 0.02
Hypothyroid	0.05 ± 0.00	0.12 ± 0.01	0.05 ± 0.00	0.24 ± 0.01	0.12 ± 0.01
Kr-vs-kp	0.08 ± 0.01	0.16 ± 0.01	0.13 ± 0.00	0.08 ± 0.00	0.12 ± 0.00
Letter	0.35 ± 0.02	0.38 ± 0.00	0.39 ± 0.01	na	na
Mushroom	0.03 ± 0.00	0.04 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
Poker-lsn	0.05 ± 0.00	0.59 ± 0.00	na	na	na
Segment	0.16 ± 0.01	0.22 ± 0.01	0.19 ± 0.01	0.14 ± 0.01	0.18 ± 0.00
Sick	0.07 ± 0.00	0.10 ± 0.01	0.08 ± 0.00	0.11 ± 0.01	0.10 ± 0.00
Splice	0.26 ± 0.01	0.11 ± 0.00	0.25 ± 0.01	na	0.19 ± 0.00
Waveform5000	0.59 ± 0.02	0.22 ± 0.01	0.52 ± 0.00	na	0.26 ± 0.01

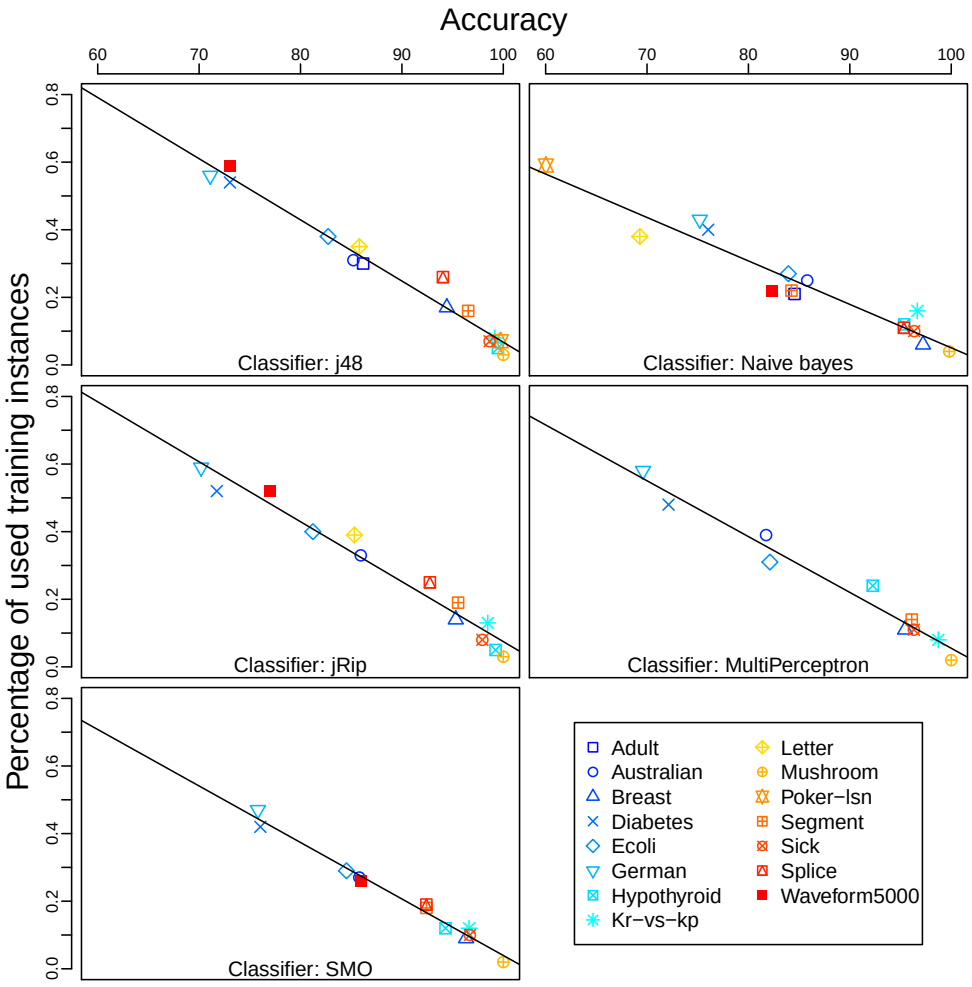


Figure 1. Correlation between accuracy and percentage of used training examples when windowing. J48 $= -0.98$, NB $= -0.96$, jRip $= -0.98$, MP $= -0.98$, and SMO $= -0.99$. In general, the models with higher accuracy use less examples to be induced.

Large datasets such as as Adult, Letter, Poker-Lsn, Splice, and Waveform5000 did not finish on reasonable time when using jRip, Multilayer-Perceptron and SMO, with and without windowing. In such cases, results are reported as not available (na). This might be solved by running the experiments in a real cluster of 8 nodes, instead of simulating the sites in a single machine, as done here, but it is not relevant for the purposes of this work. In the following results, Poker-lsn dataset was excluded because the cross-validations runs do not finish on a reasonable time, this might be solved with more computational power. The results were kept this way because they illustrate that some classifiers exhibit a computational cost which precludes convergence.

3.2. Samples Properties

For each dataset considered in this work, Table 5 shows some properties of the samples obtained by the following methods: windowing, as described before; the Full-Dataset under a 10-folds cross-validation (90% of all available data); and the random, stratified, and balanced samplings. Properties include the size of the sample in terms of the number of instances; the standard deviation of the class distribution (*St.Dv.C.D.*); and two measures of similarity between the samples and the original dataset: The Kullback–Leibler divergence and the metric sim_1. With the exception of Full-Dataset, the size of the rest of the samples is determined by the windowing method and its autostop method. For the sake of fairness, windowing is executed first and the size of the sample obtained in this way is

adopted for the rest of the sampling methods. Reductions in the size of the training set are as big as 97% of the available data (Hypothyroid).

According to Kullback–Leibler Divergence, windowing is the method that skews more the original class distribution in non-balanced datasets. It is also observed that the class distribution on the windows is more balanced, and its effectiveness probably depends on the number of available examples for the minority classes. For instance, Full-Dataset shows an unbalanced class distribution ($St.Dv.C.D. = 0.449$) in Hypothyroid, while windowing got a coefficient of 0.293. Windowing can not completely balance the number of examples per class since the percentage of the available examples for the minority classes are around of 5%. The random sampling, the Full-Dataset, and the stratified sampling do not tend to modify the class distribution. However, it does not seem to be a correlation between this coefficient and the obtained accuracy.

Full-Dataset is, without surprise, the sample that gathers more attribute/values pairs from the original data, since it uses 90% of the available data. It is included in the results exclusively for comparison with the rest of the sampling methods. Table 5 also show that windowing tends to collect more information content in most of the datasets compared with all the sampling, this is probably result of the heuristic nature of windowing. There are some datasets, like Breast and German, where all the techniques have one as the measured value of $Sim1$. Unfortunately, as in the previous case, this notion of similarity neither seems to correlate with the observed accuracy, for instance, as mentioned, for Breast and German all the sampling methods gathers all the original pairs attribute-value ($Sim_1 = 1.0$), but while the accuracy obtained for Breast is around 95%, when using German it is around 71%. In concordance with these results, the window for Breast uses 17% of the available examples, while German uses 64% (Table 5).

Table 5. Samples properties.

Dataset	Method	Instances		St. Dv. C.D.		KL Div		Sim1	
Adult	Windowing	14502.840 ±	574.266	0.083 ±	0.004	0.128 ±	0.004	0.386 ±	0.012
Adult	Full-Dataset	43957.800 ±	0.402	0.369 ±	0.000	0.000 ±	0.000	0.935 ±	0.001
Adult	Random-sampling	14502.840 ±	574.266	0.374 ±	0.049	0.005 ±	0.005	0.418 ±	0.013
Adult	Stratified-sampling	14502.840 ±	574.266	0.369 ±	0.000	0.000 ±	0.000	0.418 ±	0.013
Adult	Balanced-sampling	14502.840 ±	574.266	0.000 ±	0.000	0.206 ±	0.000	0.400 ±	0.013
Australian	Windowing	215.440 ±	14.363	0.031 ±	0.020	0.017 ±	0.008	0.999 ±	0.006
Australian	Full-Dataset	621.000 ±	0.000	0.078 ±	0.001	0.000 ±	0.000	0.999 ±	0.005
Australian	Random-sampling	215.440 ±	14.363	0.080 ±	0.047	0.004 ±	0.005	0.986 ±	0.016
Australian	Stratified-sampling	215.440 ±	14.363	0.078 ±	0.004	0.000 ±	0.000	0.986 ±	0.016
Australian	Balanced-sampling	215.440 ±	14.363	0.001 ±	0.002	0.009 ±	0.000	0.987 ±	0.016
Breast	Windowing	109.210 ±	14.732	0.043 ±	0.030	0.086 ±	0.031	1.000 ±	0.000
Breast	Full-Dataset	614.700 ±	0.461	0.212 ±	0.000	0.000 ±	0.000	1.000 ±	0.000
Breast	Random-sampling	109.210 ±	14.732	0.224 ±	0.107	0.019 ±	0.017	1.000 ±	0.000
Breast	Stratified-sampling	109.210 ±	14.732	0.215 ±	0.007	0.000 ±	0.000	1.000 ±	0.000
Breast	Balanced-sampling	109.210 ±	14.732	0.003 ±	0.003	0.066 ±	0.003	1.000 ±	0.000
Diabetes	Windowing	436.260 ±	27.768	0.087 ±	0.022	0.025 ±	0.009	0.751 ±	0.028
Diabetes	Full-Dataset	691.200 ±	0.402	0.213 ±	0.001	0.000 ±	0.000	0.954 ±	0.004
Diabetes	Random-sampling	436.260 ±	27.768	0.214 ±	0.021	0.001 ±	0.001	0.763 ±	0.028
Diabetes	Stratified-sampling	436.260 ±	27.768	0.215 ±	0.002	0.000 ±	0.000	0.766 ±	0.028
Diabetes	Balanced-sampling	436.260 ±	27.768	0.001 ±	0.001	0.067 ±	0.001	0.770 ±	0.028
Ecoli	Windowing	126.640 ±	8.579	0.109 ±	0.005	0.182 ±	0.055	0.761 ±	0.026
Ecoli	Full-Dataset	302.400 ±	0.492	0.145 ±	0.000	0.001 ±	0.001	0.979 ±	0.006
Ecoli	Random-sampling	126.640 ±	8.579	0.147 ±	0.010	0.007 ±	0.010	0.763 ±	0.025
Ecoli	Stratified-sampling	126.640 ±	8.579	0.154 ±	0.004	0.013 ±	0.003	0.758 ±	0.027
Ecoli	Balanced-sampling	126.640 ±	8.579	0.099 ±	0.004	0.113 ±	0.028	0.781 ±	0.028
German	Windowing	584.750 ±	25.308	0.119 ±	0.012	0.041 ±	0.006	1.000 ±	0.000
German	Full-Dataset	900.000 ±	0.000	0.283 ±	0.000	0.000 ±	0.000	1.000 ±	0.000
German	Random-sampling	584.750 ±	25.308	0.284 ±	0.022	0.001 ±	0.001	1.000 ±	0.000
German	Stratified-sampling	584.750 ±	25.308	0.283 ±	0.001	0.000 ±	0.000	1.000 ±	0.000
German	Balanced-sampling	584.750 ±	25.308	0.055 ±	0.022	0.079 ±	0.015	1.000 ±	0.000

Continued on next page

Dataset	Method	Instances		St. Dv. C.D.		KL Div		Sim1	
Hypothyroid	Windowing	151.680 ±	9.619	0.293 ±	0.017	0.262 ±	0.047	0.428 ±	0.017
Hypothyroid	Full-Dataset	3394.800 ±	0.402	0.449 ±	0.000	0.000 ±	0.000	0.979 ±	0.005
Hypothyroid	Random-sampling	151.680 ±	9.619	0.580 ±	0.149	0.212 ±	0.103	0.387 ±	0.020
Hypothyroid	Stratified-sampling	151.680 ±	9.619	0.516 ±	0.007	0.000 ±	0.001	0.387 ±	0.013
Hypothyroid	Balanced-sampling	151.680 ±	9.619	0.191 ±	0.004	0.668 ±	0.023	0.435 ±	0.016
Kr-vs-kp	Windowing	242.550 ±	18.425	0.050 ±	0.036	0.010 ±	0.012	0.998 ±	0.004
Kr-vs-kp	Full-Dataset	2876.400 ±	0.492	0.031 ±	0.000	0.000 ±	0.000	0.999 ±	0.004
Kr-vs-kp	Random-sampling	242.550 ±	18.425	0.221 ±	0.130	0.106 ±	0.099	0.975 ±	0.013
Kr-vs-kp	Stratified-sampling	242.550 ±	18.425	0.032 ±	0.003	0.000 ±	0.000	0.977 ±	0.009
Kr-vs-kp	Balanced-sampling	242.550 ±	18.425	0.001 ±	0.001	0.001 ±	0.000	0.977 ±	0.008
Letter	Windowing	7390.450 ±	491.435	0.008 ±	0.000	0.037 ±	0.002	0.989 ±	0.006
Letter	Full-Dataset	18000.000 ±	0.000	0.001 ±	0.000	0.000 ±	0.000	0.999 ±	0.002
Letter	Random-sampling	7390.450 ±	491.435	0.007 ±	0.001	0.022 ±	0.009	0.983 ±	0.008
Letter	Stratified-sampling	7390.450 ±	491.435	0.000 ±	0.000	0.000 ±	0.000	0.985 ±	0.007
Letter	Balanced-sampling	7390.450 ±	491.435	0.001 ±	0.000	0.001 ±	0.000	0.984 ±	0.006
Mushroom	Windowing	219.490 ±	16.871	0.043 ±	0.033	0.004 ±	0.005	0.968 ±	0.021
Mushroom	Full-Dataset	7311.600 ±	0.492	0.025 ±	0.000	0.000 ±	0.000	1.000 ±	0.000
Mushroom	Random-sampling	219.490 ±	16.871	0.504 ±	0.244	2.083 ±	1.852	0.833 ±	0.072
Mushroom	Stratified-sampling	219.490 ±	16.871	0.026 ±	0.004	0.000 ±	0.000	0.903 ±	0.032
Mushroom	Balanced-sampling	219.490 ±	16.871	0.002 ±	0.002	0.001 ±	0.000	0.902 ±	0.033
Segment	Windowing	371.280 ±	27.458	0.104 ±	0.008	0.390 ±	0.076	0.279 ±	0.015
Segment	Full-Dataset	2079.000 ±	0.000	0.000 ±	0.000	0.000 ±	0.000	0.938 ±	0.003
Segment	Random-sampling	371.280 ±	27.458	0.050 ±	0.007	0.105 ±	0.144	0.310 ±	0.019
Segment	Stratified-sampling	371.280 ±	27.458	0.002 ±	0.001	0.000 ±	0.000	0.315 ±	0.018
Segment	Balanced-sampling	371.280 ±	27.458	0.002 ±	0.001	0.000 ±	0.000	0.315 ±	0.018
Sick	Windowing	264.600 ±	17.420	0.305 ±	0.028	0.233 ±	0.032	0.565 ±	0.019
Sick	Full-Dataset	3394.800 ±	0.402	0.621 ±	0.000	0.000 ±	0.000	0.979 ±	0.005
Sick	Random-sampling	264.600 ±	17.420	0.623 ±	0.066	0.015 ±	0.014	0.483 ±	0.018
Sick	Stratified-sampling	264.600 ±	17.420	0.623 ±	0.002	0.000 ±	0.000	0.483 ±	0.014
Sick	Balanced-sampling	264.600 ±	17.420	0.002 ±	0.001	0.665 ±	0.002	0.495 ±	0.014
Splice	Windowing	835.300 ±	29.689	0.072 ±	0.011	0.036 ±	0.009	0.969 ±	0.043
Splice	Full-Dataset	2871.000 ±	0.000	0.169 ±	0.047	0.000 ±	0.000	0.987 ±	0.034
Splice	Random-sampling	835.300 ±	29.689	0.161 ±	0.000	0.014 ±	0.013	0.890 ±	0.060
Splice	Stratified-sampling	835.300 ±	29.689	0.161 ±	0.001	0.000 ±	0.000	0.862 ±	0.036
Splice	Balanced-sampling	835.300 ±	29.689	0.001 ±	0.001	0.104 ±	0.001	0.871 ±	0.046
Waveform-5000	Windowing	3263.590 ±	330.000	0.006 ±	0.004	0.000 ±	0.000	0.940 ±	0.018
Waveform-5000	Full-Dataset	4500.000 ±	0.000	0.004 ±	0.000	0.000 ±	0.000	0.983 ±	0.001
Waveform-5000	Random-sampling	3263.590 ±	330.000	0.018 ±	0.010	0.002 ±	0.002	0.932 ±	0.019
Waveform-5000	Stratified-sampling	3263.590 ±	330.000	0.004 ±	0.000	0.000 ±	0.000	0.932 ±	0.019
Waveform-5000	Balanced-sampling	3263.590 ±	330.000	0.000 ±	0.000	0.000 ±	0.000	0.932 ±	0.019

3.3. Model Complexity and Data Compression

Table 6 shows the results for the MDL, calculated using the test dataset. Respecting the number of bits required to encode a tree ($L(H)$), Windowing and Full-Dataset tend to induce more complex models, i.e, trees with more nodes. This is probably because windowing favors the search for more difficult patterns in the set of available instances, which require more complex models to be expressed. Respecting the number of bits required to encode the test data, given the induced decision tree, ($L(D|H)$) a better compression is achieved using windowing and Full-Dataset than when using the traditional samplings. Big differences in data compression using windowing are exhibit in datasets like Mushroom, Segment, and Waveform-5000. One possible explanation for this is that instances gathered by sampling techniques do not capture the data nature because of their random selection and the small number of instances in the sample.

The sum of the former metrics, the MDL, reports bigger models in most of the datasets when using windowing and Full-Dataset. This result does not represent an advantage, but properties such as the predictive performance also play an important role in model selection.

Table 6. Model complexity and test data compression.

Dataset	Method	L(H)		L(D | H)		MDL	
Adult	Windowing	1361.599 ±	465.850	2366.019 ±	59.709	3727.618 ±	483.653
Adult	Full-Dataset	2077.010 ±	282.565	2374.002 ±	49.985	4451.012 ±	270.561
Adult	Random-sampling	1009.386 ±	276.429	2420.278 ±	56.458	3429.664 ±	264.703
Adult	Stratified-sampling	1031.172 ±	181.155	2410.870 ±	49.932	3442.042 ±	186.437
Adult	Balanced-sampling	1351.736 ±	265.668	2423.024 ±	44.271	3774.759 ±	274.906
Australian	Windowing	77.299 ±	29.067	41.284 ±	6.849	118.582 ±	30.088
Australian	Full-Dataset	66.820 ±	16.934	41.044 ±	6.711	107.864 ±	17.430
Australian	Random-sampling	45.151 ±	18.592	41.820 ±	6.916	86.971 ±	19.120
Australian	Stratified-sampling	50.313 ±	22.016	41.836 ±	6.776	92.149 ±	21.220
Australian	Balanced-sampling	44.603 ±	22.878	42.327 ±	6.764	86.929 ±	22.830
Breast	Windowing	46.541 ±	13.199	25.904 ±	4.584	72.445 ±	12.435
Breast	Full-Dataset	58.757 ±	7.942	25.338 ±	5.280	84.095 ±	8.195
Breast	Random-sampling	22.301 ±	6.555	29.008 ±	7.229	51.309 ±	7.316
Breast	Stratified-sampling	23.991 ±	6.915	28.631 ±	6.720	52.622 ±	8.350
Breast	Balanced-sampling	22.767 ±	7.801	28.191 ±	5.710	50.959 ±	8.137
Diabetes	Windowing	59.000 ±	37.207	65.437 ±	5.227	124.437 ±	37.477
Diabetes	Full-Dataset	126.620 ±	46.019	64.383 ±	5.161	191.003 ±	45.988
Diabetes	Random-sampling	95.960 ±	38.989	65.674 ±	4.884	161.634 ±	39.119
Diabetes	Stratified-sampling	94.940 ±	39.261	64.354 ±	5.965	159.294 ±	39.505
Diabetes	Balanced-sampling	104.840 ±	36.621	65.263 ±	5.003	170.103 ±	36.829
Ecoli	Windowing	99.328 ±	23.152	29.959 ±	7.767	129.287 ±	23.257
Ecoli	Full-Dataset	144.454 ±	19.804	27.648 ±	6.460	172.102 ±	18.623
Ecoli	Random-sampling	69.348 ±	16.853	33.969 ±	9.853	103.317 ±	15.614
Ecoli	Stratified-sampling	65.678 ±	16.214	34.174 ±	10.710	99.852 ±	16.457
Ecoli	Balanced-sampling	83.869 ±	20.904	30.357 ±	7.087	114.226 ±	20.376
German	Windowing	315.252 ±	60.182	82.866 ±	5.220	398.118 ±	60.077
German	Full-Dataset	287.566 ±	54.049	83.857 ±	5.339	371.423 ±	53.413
German	Random-sampling	211.627 ±	51.692	83.245 ±	5.156	294.871 ±	51.783
German	Stratified-sampling	212.684 ±	54.545	83.006 ±	5.125	295.689 ±	53.830
German	Balanced-sampling	238.184 ±	51.813	84.412 ±	5.352	322.596 ±	51.356
Hypothyroid	Windowing	84.812 ±	19.108	28.291 ±	6.449	113.102 ±	20.727
Hypothyroid	Full-Dataset	122.317 ±	10.791	27.105 ±	6.877	149.422 ±	10.562
Hypothyroid	Random-sampling	15.667 ±	15.278	189.232 ±	110.454	204.899 ±	96.402
Hypothyroid	Stratified-sampling	30.645 ±	6.465	67.493 ±	22.683	98.138 ±	22.336
Hypothyroid	Balanced-sampling	45.353 ±	10.448	61.502 ±	18.798	106.854 ±	18.199
Kr-vs-kp	Windowing	198.034 ±	14.570	69.919 ±	4.871	267.953 ±	14.944
Kr-vs-kp	Full-Dataset	219.807 ±	16.870	69.345 ±	4.277	289.152 ±	17.014
Kr-vs-kp	Random-sampling	64.438 ±	18.816	98.961 ±	21.032	163.399 ±	21.636
Kr-vs-kp	Stratified-sampling	72.664 ±	18.341	92.724 ±	15.119	165.388 ±	15.947
Kr-vs-kp	Balanced-sampling	73.848 ±	18.721	91.842 ±	14.262	165.690 ±	15.840
Letter	Windowing	11862.644 ±	473.112	1248.697 ±	64.017	13111.341 ±	453.031
Letter	Full-Dataset	12431.372 ±	180.896	1165.793 ±	38.869	13597.165 ±	182.617
Letter	Random-sampling	7020.909 ±	385.222	1473.635 ±	81.356	8494.544 ±	358.576
Letter	Stratified-sampling	7102.767 ±	358.000	1461.702 ±	80.161	8564.469 ±	328.131
Letter	Balanced-sampling	7126.843 ±	381.507	1449.106 ±	76.567	8575.949 ±	354.232
Mushroom	Windowing	79.249 ±	7.033	76.881 ±	4.163	156.130 ±	7.189
Mushroom	Full-Dataset	77.237 ±	0.600	79.510 ±	1.744	156.747 ±	1.810
Mushroom	Random-sampling	18.228 ±	19.552	461.838 ±	353.124	480.066 ±	337.153
Mushroom	Stratified-sampling	31.126 ±	14.101	114.606 ±	23.525	145.732 ±	20.201
Mushroom	Balanced-sampling	31.879 ±	15.063	113.501 ±	22.427	145.380 ±	17.422
Segment	Windowing	348.723 ±	34.369	81.656 ±	10.719	430.379 ±	33.528
Segment	Full-Dataset	365.928 ±	22.569	79.045 ±	9.609	444.973 ±	22.295
Segment	Random-sampling	142.987 ±	22.538	135.754 ±	31.843	278.741 ±	31.578
Segment	Stratified-sampling	142.715 ±	18.438	126.640 ±	24.516	269.356 ±	26.762
Segment	Balanced-sampling	141.267 ±	17.852	127.325 ±	23.254	268.591 ±	26.010
Sick	Windowing	170.530 ±	26.600	50.476 ±	8.212	221.005 ±	26.977
Sick	Full-Dataset	182.701 ±	22.491	42.346 ±	7.910	225.047 ±	20.038
Sick	Random-sampling	21.786 ±	16.605	80.715 ±	38.277	102.501 ±	24.810
Sick	Stratified-sampling	31.126 ±	6.768	55.199 ±	13.736	86.325 ±	15.387

Continued on next page

Dataset	Method	L(H)		L(D|H)		MDL	
Sick	Balanced-sampling	57.996 ±	17.446	60.045 ±	9.531	118.040 ±	18.444
Splice	Windowing	725.951 ±	53.364	181.187 ±	11.871	907.139 ±	53.195
Splice	Full-Dataset	745.146 ±	51.142	179.689 ±	11.014	924.834 ±	52.532
Splice	Random-sampling	425.144 ±	52.153	187.097 ±	21.631	612.240 ±	47.209
Splice	Stratified-sampling	443.339 ±	51.337	188.061 ±	19.286	631.400 ±	48.312
Splice	Balanced-sampling	419.763 ±	41.676	188.473 ±	20.593	608.236 ±	40.687
Waveform-5000	Windowing	2418.668 ±	215.760	363.799 ±	56.499	2782.467 ±	224.433
Waveform-5000	Full-Dataset	2615.956 ±	94.305	415.810 ±	20.601	3031.766 ±	92.381
Waveform-5000	Random-sampling	1957.647 ±	203.398	413.447 ±	24.548	2371.094 ±	202.636
Waveform-5000	Stratified-sampling	1957.202 ±	199.174	417.104 ±	26.348	2374.306 ±	196.151
Waveform-5000	Balanced-sampling	1966.554 ±	193.650	417.152 ±	28.133	2383.706 ±	190.987

3.4. Predictive Performance

Table 7 shows the predictive performance in terms of accuracy and the AUC. Even though the random, stratified and balanced samplings usually induce simpler models, the decision trees do not seem to be more general than their windowing and Full-Dataset counterparts. In other words, the predictive ability of decision trees induced with the traditional samplings are, most of the time, lower than the models induced using windowing and Full-Dataset. Models induced with windowing have the same accuracy as those obtained by Full-Dataset and, sometimes, they even show a higher accuracy, e.g., waveform-500. In terms of AUC, windowing and Full-Dataset were the best samples, but the balanced sampling is pretty close to their performance.

Table 7. Predictive performance.

Dataset	Method	Test Acc		Test AUC	
Adult	Windowing	86.355 ±	0.889	78.227 ±	1.161
Adult	Full-Dataset	86.074 ±	0.390	77.080 ±	0.823
Adult	Random-sampling	85.516 ±	0.423	76.131 ±	2.021
Adult	Stratified-sampling	85.677 ±	0.401	76.680 ±	0.885
Adult	Balanced-sampling	80.489 ±	0.722	81.956 ±	0.580
Australian	Windowing	85.710 ±	4.355	85.471 ±	4.411
Australian	Full-Dataset	86.536 ±	3.969	86.239 ±	4.041
Australian	Random-sampling	85.101 ±	4.375	84.849 ±	4.517
Australian	Stratified-sampling	85.391 ±	4.164	85.142 ±	4.266
Australian	Balanced-sampling	85.536 ±	3.925	85.584 ±	3.854
Breast	Windowing	94.829 ±	2.804	94.368 ±	3.117
Breast	Full-Dataset	95.533 ±	2.674	95.058 ±	2.830
Breast	Random-sampling	92.696 ±	3.821	91.687 ±	4.739
Breast	Stratified-sampling	92.783 ±	3.485	91.956 ±	3.982
Breast	Balanced-sampling	92.433 ±	3.558	92.301 ±	3.627
Diabetes	Windowing	74.161 ±	4.864	70.041 ±	5.654
Diabetes	Full-Dataset	74.756 ±	4.661	71.211 ±	5.027
Diabetes	Random-sampling	72.280 ±	4.520	68.602 ±	5.403
Diabetes	Stratified-sampling	73.222 ±	5.113	70.254 ±	5.721
Diabetes	Balanced-sampling	71.018 ±	5.222	71.726 ±	4.937
Ecoli	Windowing	82.777 ±	6.353	88.848 ±	4.134
Ecoli	Full-Dataset	82.822 ±	5.467	88.873 ±	3.567
Ecoli	Random-sampling	80.059 ±	6.268	86.924 ±	4.218
Ecoli	Stratified-sampling	79.586 ±	6.227	86.721 ±	4.113
Ecoli	Balanced-sampling	79.405 ±	6.360	86.981 ±	4.034
German	Windowing	71.660 ±	4.608	63.119 ±	5.518
German	Full-Dataset	71.300 ±	3.765	62.605 ±	4.388
German	Random-sampling	71.800 ±	3.782	62.867 ±	4.408
German	Stratified-sampling	71.640 ±	3.799	62.857 ±	4.546
German	Balanced-sampling	67.820 ±	4.448	66.833 ±	4.014
Hypothyroid	Windowing	99.483 ±	0.346	98.880 ±	1.204
Hypothyroid	Full-Dataset	99.528 ±	0.353	98.871 ±	1.259
Hypothyroid	Random-sampling	94.340 ±	2.524	70.634 ±	23.378
Hypothyroid	Stratified-sampling	96.877 ±	1.652	94.594 ±	4.769

Continued on next page

Dataset	Method	Test Acc		Test AUC	
Hypothyroid	Balanced-sampling	96.236 ±	1.831	97.598 ±	1.421
Kr-vs-kp	Windowing	99.302 ±	0.583	99.294 ±	0.594
Kr-vs-kp	Full-Dataset	99.415 ±	0.433	99.412 ±	0.433
Kr-vs-kp	Random-sampling	94.171 ±	2.959	94.139 ±	3.061
Kr-vs-kp	Stratified-sampling	94.956 ±	1.766	94.956 ±	1.802
Kr-vs-kp	Balanced-sampling	94.984 ±	1.727	94.996 ±	1.756
Letter	Windowing	87.161 ±	2.074	93.324 ±	1.078
Letter	Full-Dataset	87.943 ±	0.720	93.731 ±	0.375
Letter	Random-sampling	82.216 ±	1.006	90.753 ±	0.523
Letter	Stratified-sampling	82.376 ±	1.148	90.836 ±	0.597
Letter	Balanced-sampling	82.430 ±	1.160	90.864 ±	0.603
Mushroom	Windowing	100.000 ±	0.000	100.000 ±	0.000
Mushroom	Full-Dataset	100.000 ±	0.000	100.000 ±	0.000
Mushroom	Random-sampling	73.746 ±	23.610	73.625 ±	23.684
Mushroom	Stratified-sampling	98.367 ±	0.813	98.312 ±	0.831
Mushroom	Balanced-sampling	98.424 ±	0.819	98.376 ±	0.831
Segment	Windowing	96.329 ±	1.655	97.859 ±	0.965
Segment	Full-Dataset	96.710 ±	1.335	98.081 ±	0.779
Segment	Random-sampling	90.719 ±	3.181	94.586 ±	1.855
Segment	Stratified-sampling	91.515 ±	2.074	95.051 ±	1.210
Segment	Balanced-sampling	91.455 ±	1.984	95.015 ±	1.157
Sick	Windowing	98.688 ±	0.640	93.667 ±	3.370
Sick	Full-Dataset	98.741 ±	0.523	93.662 ±	3.323
Sick	Random-sampling	96.193 ±	1.887	75.662 ±	19.843
Sick	Stratified-sampling	97.301 ±	1.051	86.908 ±	6.166
Sick	Balanced-sampling	94.785 ±	1.855	94.812 ±	2.641
Splice	Windowing	94.132 ±	1.682	95.626 ±	1.344
Splice	Full-Dataset	94.216 ±	1.474	95.723 ±	1.125
Splice	Random-sampling	89.997 ±	2.226	92.370 ±	1.951
Splice	Stratified-sampling	90.339 ±	1.973	92.757 ±	1.572
Splice	Balanced-sampling	89.846 ±	2.199	92.902 ±	1.570
Waveform-5000	Windowing	83.802 ±	9.864	87.848 ±	7.402
Waveform-5000	Full-Dataset	75.202 ±	1.989	81.396 ±	1.493
Waveform-5000	Random-sampling	75.046 ±	2.159	81.279 ±	1.619
Waveform-5000	Stratified-sampling	75.252 ±	1.981	81.431 ±	1.487
Waveform-5000	Balanced-sampling	75.514 ±	2.143	81.628 ±	1.609

3.5. Statistical Tests

The figures in this section visualize the results of the post-hoc Nemenyi test for the metrics previously shown in Tables 5, 6 and 7. This compact, information-dense visualization, called as Critical Difference diagram, consists on a main axis where the average rank of each methods is plotted along with a line that represents the Critical Difference (CD). Methods separated by a distance shorter than the CD are statistically indistinguishable, i.e., the evidence is not sufficient to conclude whether they have a similar performance and are connected by a black line. In contrast, methods separated by a distance larger than the CD have a statistically significant difference in performance. The best performing methods are those with lower rank values shown on the left of the figure.

Figure 2 shows the results for the number of bits required to encode the induced models ($L(H)$) presented in Table 6. The groups of connected algorithms are not significantly different. In this case, the complexity of the models induced using windowing does not show significant differences with the complexity of the models induced using the Full-Dataset or balanced sampling.

Figure 3 shows the results in terms of data compression given the decision tree ($L(D|H)$). If the compressibility provided by the models is verified on a stratified sample of unseen data, windowing and Full-Dataset tend to compress significantly better compared to traditional sampling methods. However, windowing tends to generate more complex models probably because its heuristic behavior enables the seek for more difficult patterns in the data.

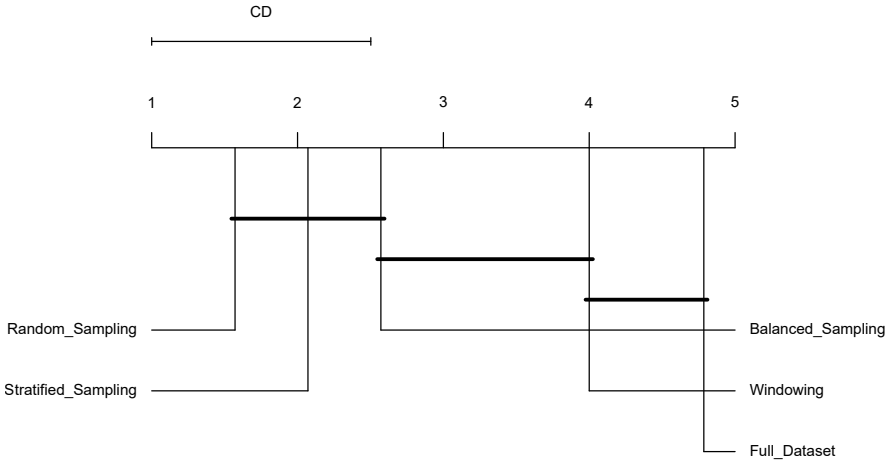


Figure 2. Demšar test regarding the required bits to encode trees, $L(H)$.

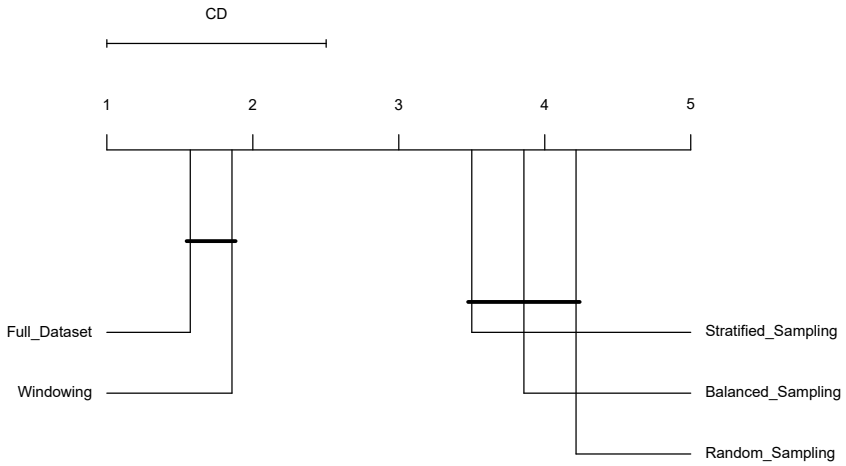


Figure 3. Demšar test regarding the required bits to encode the test data given the decision tree, $L(D|H)$.

Figure 4 shows the results in terms of MDL in the test set. Windowing and Full-Dataset do not show significant differences, nor they are statistically different to the traditional sampling methods. That is, that the induced decision trees generally need the same number of bits to be represented.

Figure 5 shows the results for accuracy. Windowing performs very well, being almost as accurate as Full-Dataset without significant differences. Both methods are strictly better than the random, balanced, and stratified samplings. When considering the AUC in Figure 6, results are very similar but the balanced sampling does not show significant differences with windowing and the Full-Dataset. Recall that both, windowing and balanced sampling, tend to balance the class distribution of the instances.

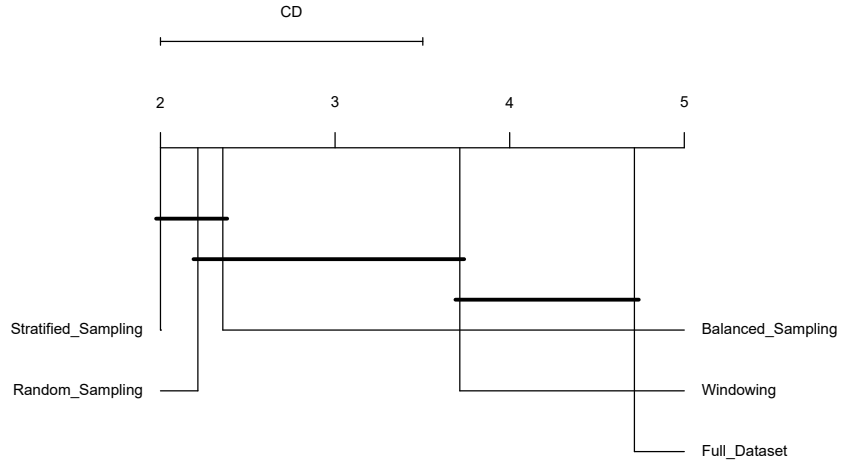


Figure 4. Demšar test regarding the MDL computed on the test dataset.

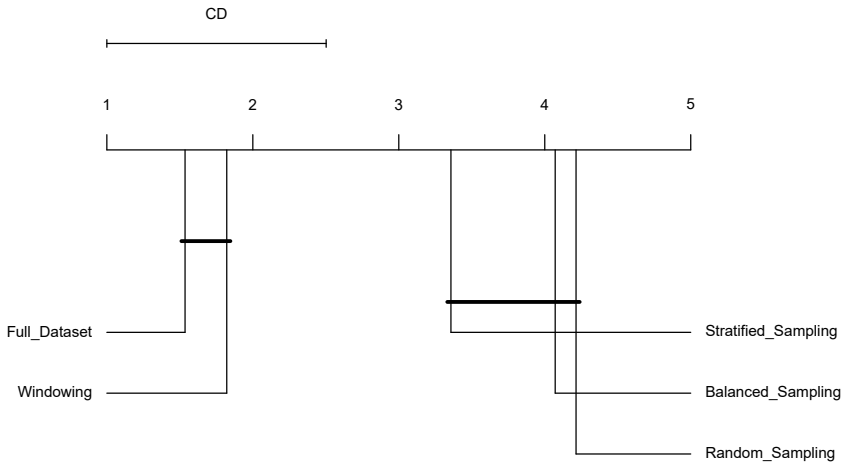


Figure 5. Demšar test regarding the accuracy over the test dataset.

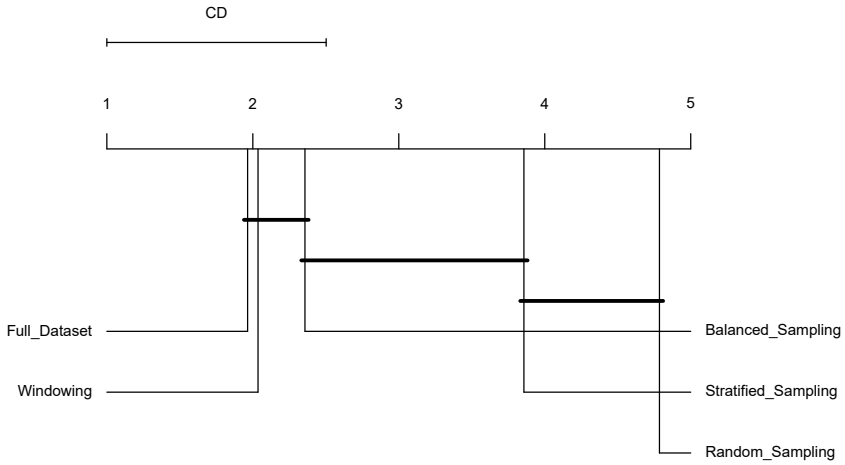


Figure 6. Demšar test regarding the AUC over the test dataset.

In terms of class distribution (Figure 7), windowing is known to be the method that tends to skew the distribution the most, given that the counter examples added to the window in each iteration of this algorithm belong most probably to the current minority class. As expected, the balanced and the

random sampling methods also skew the class distribution showing no significant differences with windowing. According to the percentage of attribute-value pairs given by Sim_1 (Figure 8), windowing and the traditional sampling methods cannot obtain the full set of attribute-value pairs included in the original dataset. Despite this, windowing is still very competent when it comes to prediction.

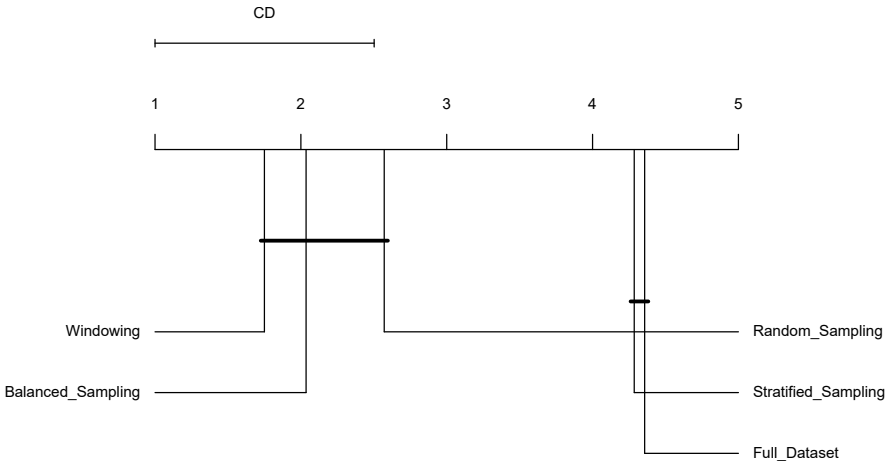


Figure 7. Demšar test regarding the Kullback–Leibler Divergence.

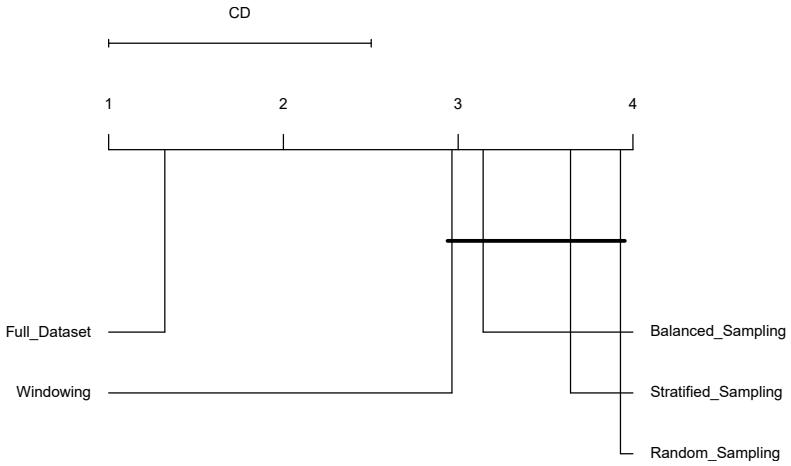


Figure 8. Demšar test regarding Sim1.

4. Conclusions

The generalization of the behavior of windowing beyond decision trees and the J48 algorithm has been corroborated. Independently of the inductive method used with windowing, high accuracies correlate with aggressive samplings up to 3% of the original datasets. This result motivates the study of the properties of the samples and models proposed in this work. Unfortunately, the Kullback–Leibler divergence and sim_1 do not seem to correlate with accuracy, although the first one is indicative of the balancing effect performed by windowing. MDL provided useful information in the sense that, although all methods generate models of similar complexity, it is important to identify which component of the MDL is more relevant in each case. For example, less complex decision trees, as those induced by random, balanced and stratified samplings, are more general but less accurate. In contrast, decision trees with better data compression, such as those induced using windowing and Full-Dataset, tend to be larger but more accurate. The key factor that makes the difference is the significant reduction of instances for induction. Recall that determining the size of the samples is

done automatically in windowing, based on the auto-stop condition of this method. When using traditional sampling methods the size must be figured out by the user of the technique. To the best of our knowledge, this is the first comparative study of windowing in this respect. This work suggests future lines of research on windowing, including:

1. Adopting metrics for detecting relevant, noisy, and redundant instances to enhance the quality and size of the obtained samples, in order to improve the performance of the obtained models. Maillo *et al.* [30] review multiple metrics to describe redundancy, complexity, and density of a problem and also propose two data big metrics. These kind of metrics may be helpful to select instances that provides quality information.
2. Studying the evolution of windows over time can offer more insights about the behavior of windowing. The main difficulty here is adapting some of the used metrics, e.g., MDL, to be used with models that are not decision trees.
3. Dealing with datasets of higher dimensions. Melgoza-Gutiérrez *et al.* [31] propose an agent & artifacts-based method to distribute vertical partitions of datasets and deal with the growing time complexity when datasets have a high number of attributes. It is expected that the achieved understanding on windowing contributes to combine these approaches.
4. Applying windowing to real problems. Limón *et al.* [10] applies windowing to the segmentation of colposcopic images presenting possible precancerous cervical lesions. Windowing is exploited here to distribute the computational cost of processing a dataset of 1.4×10^6 instances and 30 attributes. The exploitation of windowing to cope with learning problems of distributed nature is to be explored.

Author Contributions: Conceptualization, D.M.-G. and A.G.-H.; methodology, D.M.-G., A.G.-H. and N.C.-R.; software, A.G.-H., X.L. and D.M.-G.; validation, A.G.-H., N.C.-R., X.L. and F.G.; formal analysis, D.M.-G. and A.G.-H.; investigation, A.G.-H. and D.M.-G.; resources, X.L.; writing—original draft preparation, D.M.-G.; writing—review and editing, A.G.-H., N.C.-R., X.L. and F.G.; visualization, D.M.-G.; project administration, A.G.-H. All authors have read and agree to the published version of the manuscript.

Funding: The first author was funded by a scholarship from Consejo Nacional de Ciencia y Tecnología (CONACYT), Mexico, CVU:895160. The last author was supported by project RTI2018-095820-B-I00 (MCIU/AEI/FEDER, UE).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Results of Accuracy without Using Windowing

Table A1. Average accuracy without using windowing under a 10-fold cross validation (na = not available).

	j48	NB	jRip	MP	SMO
Adult	85.98 ± 0.28	83.24 ± 0.19	84.65 ± 0.16	na	na
Australian	87.10 ± 0.65	85.45 ± 1.57	84.44 ± 1.78	83.10 ± 1.28	86.71 ± 1.43
Breast	96.16 ± 0.38	97.84 ± 0.51	95.03 ± 0.89	96.84 ± 0.77	96.67 ± 0.40
Credit-g	73.59 ± 2.11	75.59 ± 1.04	73.45 ± 1.96	73.10 ± 0.72	76.66 ± 2.87
Diabetes	72.95 ± 0.77	75.83 ± 1.17	78.27 ± 1.81	74.51 ± 1.46	78.02 ± 1.79
Ecoli	84.44 ± 1.32	83.5 ± 1.64	82.25 ± 3.11	83.69 ± 1.44	83.93 ± 1.31
German	73.89 ± 1.59	76.94 ± 2.29	70.06 ± 0.90	70.26 ± 0.96	74.55 ± 1.76
Hypothyroid	99.48 ± 0.20	95.72 ± 0.68	99.60 ± 0.15	94.38 ± 0.25	94.01 ± 0.48
Kr-vs-kp	99.31 ± 0.06	87.68 ± 0.43	99.37 ± 0.29	99.06 ± 0.13	96.67 ± 0.37
Letter	87.81 ± 0.10	64.33 ± 0.28	86.34 ± 0.22	na	na
Mushroom	100.0 ± 0.00	95.9 ± 0.32	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
Poker-lsn	99.79 ± 0.00	59.33 ± 0.03	na	na	na
Segment	96.02 ± 0.29	79.95 ± 0.69	95.25 ± 0.52	95.61 ± 0.91	92.97 ± 0.36
Sick	98.88 ± 0.29	93.13 ± 0.43	98.19 ± 0.22	95.81 ± 0.45	93.70 ± 0.56
Splice	93.81 ± 0.39	95.05 ± 0.36	94.19 ± 0.27	na	93.46 ± 0.48
Waveform5000	75.58 ± 0.37	80.25 ± 0.33	79.54 ± 0.37	na	86.81 ± 0.21

References

1. Quinlan, J.R. Induction over large data bases. Technical Report STAN-CS-79-739, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, CA, USA, 1979.
2. Quinlan, J.R. Learning efficient classification procedures and their application to chess en games. In *Machine Learning*; Michalski, R.S., Carbonell, J.G., Mitchell, T.M., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1983; Volume I, Chapter 15, pp. 463–482. doi:10.1016/B978-0-08-051054-5.50019-4.
3. Quinlan, J.R. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106.
4. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: San Mateo, CA, USA, 1993; Volume 1.
5. Quinlan, J.R. Improved Use of Continuous Attributes in C4.5. *J. Artif. Intell. Res.* **1996**, *4*, 77–90.
6. Wirth, J.; Catlett, J. Experiments on the Costs and Benefits of Windowing in ID3. In Proceedings of the Fifth International Conference on Machine Learning, Ann Arbor, MI, USA, 12–14 June 1988; Laird, J.E., Ed.; Morgan Kaufmann: San Mateo, CA, USA, 1988; pp. 87–99.
7. Fürnkranz, J. Integrative windowing. *J. Artif. Intell. Res.* **1998**, *8*, 129–164.
8. Quinlan, J.R. Learning Logical Definitions from Relations. *Mach. Learn.* **1990**, *5*, 239–266.
9. Limón, X.; Guerra-Hernández, A.; Cruz-Ramírez, N.; Grimaldo, F. Modeling and implementing distributed data mining strategies in JaCa-DDM. *Knowl. Inf. Syst.* **2019**, *60*, 99–143. doi:10.1007/s10115-018-1222-x.
10. Limón, X.; Guerra-Hernández, A.; Cruz-Ramírez, N.; Acosta-Mesa, H.G.; Grimaldo, F. A Windowing Strategy for Distributed Data Mining Optimized through GPUs. *Pattern Recognit. Lett.* **2017**, *93*, 23–30. doi:10.1016/j.patrec.2016.11.006.
11. Witten, I.H.; Frank, E.; Hall, M.A. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann Publishers: Burlington, MA, USA, 2011.
12. Dua, D.; Graff, C. UCI Machine Learning Repository, 2017. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 29 June 2020).
13. Bifet, A.; Holmes, G.; Kirkby, R.; Pfahringer, B. MOA: Massive Online Analysis. *J. Mach. Learn. Res.* **2010**, *11*, 1601–1604.
14. John, G.H.; Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–20 August 1995; Morgan Kaufmann: San Mateo, CA, USA, 1995; pp. 338–345.
15. Cohen, W.W. Fast Effective Rule Induction. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; Morgan Kaufmann: San Francisco, CA, USA, 1995; pp. 115–123.

16. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J., Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*; MIT Press: Cambridge, MA, USA, 1986; pp. 318–362.
17. Platt, J. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In *Advances in Kernel Methods: Support Vector Learning*; Schoelkopf, B., Burges, C., Smola, A., Eds.; MIT Press: Cambridge, MA, USA, 1998.
18. Sokolova, M.; Lapalme, G. A Systematic Analysis of Performance Measures for Classification Tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. doi:10.1016/j.ipm.2009.03.002.
19. Provost, F.; Domingos, P. Well-Trained PETs: Improving Probability Estimation Trees (2000). Available online: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.309 (accessed on 29 June 2020).
20. Rissanen, J. Stochastic Complexity and Modeling. *Ann. Stat.* **1986**, *14*, 1080–1100. doi:10.1214/aos/1176350051.
21. Quinlan, J.R.; Rivest, R.L. Inferring decision trees using the minimum description length principle. *Inf. Comput.* **1989**, *80*, 227–248.
22. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86.
23. Zhang, S.; Zhang, C.; Wu, X. *Knowledge Discovery in Multiple Databases*; Springer-Verlag London, Limited: London, UK, 2004.
24. Ros, F.; Guillaume, S. *Sampling Techniques for Supervised or Unsupervised Tasks*; Springer: Cham, Switzerland, 2019. doi:10.1007/978-3-030-29349-9.
25. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
26. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701.
27. Friedman, M. A Comparison of Alternative Tests of Significance for the Problem of *m* Rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. doi:10.1214/aoms/1177731944.
28. Zar, J.H. *Biostatistical Analysis (5th Edition)*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2007.
29. Iman, R.L.; Davenport, J.M. Approximations of the critical region of the fbietkan statistic. *Commun. Stat. Theory Methods* **1980**, *9*, 571–595. doi:10.1080/03610928008827904.
30. Maillo, J.; Triguero, I.; Herrera, F. Redundancy and Complexity Metrics for Big Data Classification: Towards Smart Data. *IEEE Access* **2020**, *8*, 87918–87928.
31. Melgoza-Gutiérrez, J.; Guerra-Hernández, A.; Cruz-Ramírez, N. Collaborative Data Mining on a BDI Multi-Agent System over Vertically Partitioned Data. In Proceedings of the 13th Mexican International Conference on Artificial Intelligence, Tuxtla Gutiérrez, Mexico, 16–22 November 2014; Gelbukh, A., Castro-Espinoza, F., Galicia-Haro, S.N., Eds.; IEEE Computer Society: Los Alamitos, CA, USA, 2014; pp. 215–220.

Bibliography

[1] Robert J. Schalkoff. *Artificial Intelligence Engine*. USA: McGraw-Hill, Inc., 1990 (cited on page 1).

[2] Ian H Witten, Eibe Frank, and Mark A Hall. *Data Mining: Practical Machine Learning Toools and Techniques*. Burlington, MA., USA: Morgan Kaufmann Publishers, 2011 (cited on pages 1, 22).

[3] John Ross Quinlan. *Induction over large data bases*. Tech. rep. STAN-CS-79-739. Stanford, CA, USA: Computer Science Department, School of Humanities and Sciences, Stanford University, May 1979 (cited on pages 1, 4).

[4] Johannes Fürnkranz. 'More Efficient Windowing'. In: *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA*. Ed. by Benjamin Kuipers and Bonnie L. Webber. AAAI Press / The MIT Press, 1997, pp. 509–514 (cited on pages 2, 4, 6).

[5] Johannes Fürnkranz. 'Noise-Tolerant Windowing'. In: *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29, 1997, 2 Volumes*. Morgan Kaufmann, 1997, pp. 852–859 (cited on pages 2, 6).

[6] Johannes Fürnkranz. 'Integrative windowing'. In: *Journal of Artificial Intelligence Research* 8 (1998), pp. 129–164 (cited on pages 2, 3, 6, 25, 35, 55).

[7] Xavier Limón et al. 'A Windowing strategy for Distributed Data Mining optimized through GPUs'. In: *Pattern Recognition Letters* 93.Suplement C (July 2017), pp. 23–30. DOI: 10.1016/j.patrec.2016.11.006 (cited on pages 3, 7, 24, 56).

[8] Xavier Limón et al. 'Modeling and implementing distributed data mining strategies in JaCa-DDM'. In: *Knowledge and Information Systems* 60.1 (2019), pp. 99–143. DOI: 10.1007/s10115-018-1222-x (cited on pages 3, 7, 21, 22, 26, 28, 31).

[9] Jarryl Wirth and Jason Catlett. 'Experiments on the Costs and Benefits of Windowing in ID3'. In: *Machine Learning, Proceedings of the Fifth International Conference on Machine Learning, Ann Arbor, Michigan, USA, June 12-14, 1988*. Ed. by John E. Laird. Morgan Kaufmann, 1988, pp. 87–99 (cited on pages 4, 5).

[10] John Ross Quinlan. 'Learning efficient classification procedures and their application to chess en games'. In: *Machine Learning*. Ed. by Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell. Vol. I. San Francisco (CA): Morgan Kaufmann, 1983. Chap. 15, pp. 463–482. DOI: https://doi.org/10.1016/B978-0-08-051054-5.50019-4 (cited on page 5).

[11] Jason Catlett. 'Megainduction: A Test Flight'. In: *Proceedings of the Eighth International Workshop (ML91), Northwestern University, Evanston, Illinois, USA*. Ed. by Lawrence Birnbaum and Gregg Collins. Morgan Kaufmann, 1991, pp. 596–599. DOI: 10.1016/b978-1-55860-200-7.50121-5 (cited on page 5).

[12] John Ross Quinlan. *C4. 5: programs for machine learning*. Vol. 1. San Mateo, CA., USA: Morgan kaufmann, 1993 (cited on page 6).

[13] Ronald L. Rivest J. Ross Quinlan. *Inferring decision trees using the minimum description length principle*. 1989 (cited on pages 7, 27).

[14] Pedro Domingos Foster Provost. *Well-Trained PETs: Improving Probability Estimation Trees*. 2000 (cited on pages 7, 26).

[15] Shichao Zhang, Chengqi Zhang, and Xindong Wu. *Knowledge Discovery in Multiple Databases*. Advanced Information and Knowledge Processing. London, UK: Springer-Verlag London, Limited, 2004 (cited on pages 8, 25).

[16] Solomon Kullback and Richard A Leibler. 'On information and sufficiency'. In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86 (cited on pages 8, 25).

[17] Martin Møller. 'Supervised learning on large redundant training sets'. In: *International Journal of Neural Systems* 4.1 (1993), pp. 15–25 (cited on pages 8, 25).

[18] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. 'The KDD Process for Extracting Useful Knowledge from Volumes of Data'. In: *Commun. ACM* 39.11 (Nov. 1996), pp. 27–34. DOI: 10.1145/240455.240464 (cited on pages 9–11).

[19] Yanchang Zhao. *R and Data Mining: Examples and Case Studies*. Academic Press, Elsevier, Dec. 2012, p. 256 (cited on page 10).

[20] Arjun Panesar. 'What Is Machine Learning?' In: *Machine Learning and AI for Healthcare : Big Data for Improved Health Outcomes*. Berkeley, CA: Apress, 2019, pp. 75–118. DOI: 10.1007/978-1-4842-3799-1_3 (cited on page 11).

[21] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011 (cited on pages 11, 12).

[22] Marina Sokolova and Guy Lapalme. 'A systematic analysis of performance measures for classification tasks'. In: *Information processing & management* 45.4 (2009), pp. 427–437 (cited on pages 11–13, 26).

[23] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell. 'An Overview of Machine Learning'. In: *Machine Learning: An Artificial Intelligence Approach*. Ed. by R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. Berlin, Heidelberg: Springer, 1984, pp. 3–23 (cited on page 11).

[24] Li Zeng et al. 'Distributed data mining: A survey'. In: *Information Technology and Management* 13 (Dec. 2012). DOI: 10.1007/s10799-012-0124-y (cited on page 11).

[25] Bharti Suri, Manoj Kumar, et al. 'Performance Evaluation of Data Mining Techniques'. In: *Information and Communication Technology for Sustainable Development*. Springer, 2018, pp. 375–383 (cited on pages 12, 13).

[26] Tom Fawcett. 'An introduction to ROC analysis'. In: *Pattern recognition letters* 27.8 (2006), pp. 861–874 (cited on page 13).

[27] Lucas Pereira and Nuno Nunes. *A comparison of performance metrics for event classification in Non-Intrusive Load Monitoring*. 2017. DOI: 10.1109/smartgridcomm.2017.8340682 (cited on page 13).

[28] Saroj Ratnoo Pooja. 'A Comparative Study of Instance Reduction Techniques'. In: *Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM*. Citeseer. 2013 (cited on page 14).

[29] Huan Liu and Hiroshi Motoda. 'On issues of instance selection'. In: *Data Mining and Knowledge Discovery* 6.2 (2002), p. 115 (cited on page 14).

[30] J Arturo Olvera-López et al. 'A review of instance selection methods'. In: *Artificial Intelligence Review* 34.2 (2010), pp. 133–143 (cited on page 14).

[31] Karina Gibert, Miquel Sánchez–Marré, and Joaquín Izquierdo. 'A survey on pre-processing techniques: Relevant issues in the context of environmental data mining'. In: *AI Communications* 29.6 (2016), pp. 627–663 (cited on page 14).

[32] Zhi-Hua Zhou. 'Ensemble Learning'. In: *Encyclopedia of Biometrics*. Boston, MA: Springer US, 2009, pp. 270–273. DOI: 10.1007/978-0-387-73003-5_293 (cited on page 15).

[33] David Cohn, Les Atlas, and Richard Ladner. 'Improving generalization with active learning'. In: *Machine learning* 15.2 (1994), pp. 201–221 (cited on pages 15, 19).

[34] Venkateswarlu Kolluri Foster Provost. *A Survey of Methods for Scaling Up Inductive Algorithms*. 1999 (cited on pages 15, 16).

[35] Norbert Jankowski and Marek Grochowski. 'Comparison of Instances Seletion Algorithms I. Algorithms Survey'. In: *Artificial Intelligence and Soft Computing - ICAISC 2004*. Ed. by Leszek Rutkowski et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 598–603 (cited on page 16).

[36] Peter Hart. 'The condensed nearest neighbor rule (Corresp.)' In: *IEEE transactions on information theory* 14.3 (1968), pp. 515–516 (cited on page 16).

[37] G. Gates. 'The reduced nearest neighbor rule (Corresp.)' In: *IEEE Transactions on Information Theory* 18.3 (1972), pp. 431–433 (cited on page 16).

[38] David W. Aha, Dennis Kibler, and Marc K. Albert. 'Instance-Based Learning Algorithms'. In: *Mach. Learn.* 6.1 (Jan. 1991), pp. 37–66. DOI: 10.1023/A:1022689900470 (cited on page 16).

[39] RM Cameron-Jones. 'Instance selection by encoding length heuristic with random mutation hill climbing'. In: *Eighth Australian Joint Conference on Artificial Intelligence*. 1995, pp. 99–106 (cited on page 16).

[40] Leo Breiman. *Bagging Predictors*. 1994 (cited on page 17).

[41] Harris Drucker, Robert E. Schapire, and Patrice Simard. 'Improving Performance in Neural Networks Using a Boosting Algorithm'. In: *Advances in Neural Information Processing Systems 5, [NIPS Conference]*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992, pp. 42–49 (cited on page 17).

[42] Steven Simske. *Meta-Analytics: Consensus Approaches and System Patterns for Data Analysis*. Morgan Kaufmann Publishers, 2019 (cited on page 18).

[43] Pedro Domingos. *Using Partitioning to Speed Up Specific-to-General Rule Induction*. 1996 (cited on page 18).

[44] Pedro M Domingos. 'Efficient Specific-to-General Rule Induction.' In: *KDD*. 1996, pp. 319–322 (cited on page 18).

[45] David Lewis and William Gale. 'Training text classifiers by uncertainty sampling'. In: *seventeenth annual international ACM SIGIR conference on research and development in information retrieval*. 1994, pp. 3–12 (cited on page 18).

[46] Aron Culotta and Andrew McCallum. 'Reducing Labeling Effort for Structured Prediction Tasks'. In: *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*. AAAI'05. Pittsburgh, Pennsylvania: AAAI Press, 2005, pp. 746–751 (cited on page 18).

[47] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 'Active Hidden Markov Models for Information Extraction'. In: *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*. IDA '01. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 309–318 (cited on page 19).

[48] Ido Dagan and Sean P. Engelson. 'Committee-Based Sampling for Training Probabilistic Classifiers'. In: *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*. ICML'95. Tahoe City, California, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 150–157 (cited on pages 19, 20).

[49] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. 'Query by committee'. In: *Proceedings of the fifth annual workshop on Computational learning theory*. ACM. 1992, pp. 287–294 (cited on page 20).

[50] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml (cited on page 23).

[51] Albert Bifet et al. 'Moa: Massive online analysis'. In: *Journal of Machine Learning Research* 11.May (2010), pp. 1601–1604 (cited on page 23).

[52] George H. John and Pat Langley. 'Estimating Continuous Distributions in Bayesian Classifiers'. In: *Eleventh Conference on Uncertainty in Artificial Intelligence*. San Mateo: Morgan Kaufmann, 1995, pp. 338–345 (cited on page 24).

[53] William W. Cohen. 'Fast Effective Rule Induction'. In: *Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 115–123 (cited on page 24).

[54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 'Learning Internal Representations by Error Propagation'. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362 (cited on page 24).

[55] J. Platt. 'Fast Training of Support Vector Machines using Sequential Minimal Optimization'. In: *Advances in Kernel Methods - Support Vector Learning*. Ed. by B. Schoelkopf, C. Burges, and A. Smola. MIT Press, 1998 (cited on page 24).

[56] Jorma Rissanen. 'Stochastic Complexity and Modeling'. In: *The Annals of Statistics* 14 (1986), pp. 1080–1100. DOI: 10.1214/aos/1176350051 (cited on page 27).

[57] Frederic Ros and Serge Guillaume. *Sampling Techniques for Supervised or Unsupervised Tasks*. Springer, Dec. 2019 (cited on pages 27, 28).

[58] Janez Demšar. 'Statistical Comparisons of Classifiers over Multiple Data Sets'. In: *J. Mach. Learn. Res.* 7 (2006), pp. 1–30 (cited on page 28).

[59] M. Friedman. 'The use of ranks to avoid the assumption of normality implicit in the analysis of variance'. In: *Journal of the American Statistical Association* 32.200 (1937), pp. 675–701 (cited on page 28).

[60] Milton Friedman. 'A Comparison of Alternative Tests of Significance for the Problem of m Rankings'. In: *Ann. Math. Statist.* 11.1 (Mar. 1940), pp. 86–92. DOI: 10.1214/aoms/1177731944 (cited on page 28).

[61] Jerrold H. Zar. *Biostatistical Analysis (5th Edition)*. USA: Prentice-Hall, Inc., 2007 (cited on page 28).

[62] Ronald L. Iman and James M. Davenport. 'Approximations of the critical region of the fbietkan statistic'. In: *Communications in Statistics - Theory and Methods* 9.6 (1980), pp. 571–595. DOI: 10.1080/03610928008827904 (cited on page 28).

[63] J. Maillo, I. Triguero, and F. Herrera. 'Redundancy and Complexity Metrics for Big Data Classification: Towards Smart Data'. In: *IEEE Access* 8 (2020), pp. 87918–87928 (cited on page 55).

[64] Jorge Melgoza-Gutiérrez, Alejandro Guerra-Hernández, and Nicandro Cruz-Ramírez. 'Collaborative Data Mining on a BDI Multi-Agent System over Vertically Partitioned Data'. In: *13th Mexican International Conference on Artificial Intelligence: Special Session, Revised Papers*. Ed. by Alexander Gelbukh, Féix Castro-Espinoza, and Sofía N Galicia-Haro. Los Alamitos, CA, USA: IEEE Computer Society, 2014, pp. 215–220 (cited on page 55).

Alphabetical Index