Expert Systems With Applications 179 (2021) 115072

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Performance comparison of feature selection and extraction methods with random instance selection

Milad Malekipirbazari[a], Vural Aksakalli[b,*], Waleed Shafqat[b], Andrew Eberhard[b]

[a] Department of Industrial Engineering, Bilkent University, TR-06800 Bilkent, Ankara, Turkey
[b] School of Science, Royal Melbourne Institute of Technology, 124 La Trobe St, Melbourne, VIC 3000, Australia

ARTICLE INFO

Keywords:
Explainable artificial intelligence
Dimension reduction
Feature selection
Feature extraction
Instance selection
Data preprocessing

ABSTRACT

In pattern recognition, irrelevant and redundant features together with a large number of noisy instances in the underlying dataset decrease performance of trained models and make the training process considerably slower, if not practically infeasible. In order to combat this so-called curse of dimensionality, one option is to resort to feature selection (FS) methods designed to select the features that contribute the most to the performance of the model, and one other option is to utilize feature extraction (FE) methods that map the original feature space into a new space with lower dimensionality. These two methods together are called feature reduction (FR) methods. On the other hand, deploying an FR method on a dataset with massive number of instances can become a major challenge, from both memory and run time perspectives, due to the complex numerical computations involved in the process. The research question we consider in this study is rather a simple, yet novel one: do these FR methods really need the whole set of instances (WSI) available for the best performance, or can we achieve similar performance levels with selecting a much smaller random subset of WSI prior to deploying an FR method? In this work, we provide empirical evidence based on comprehensive computational experiments that the answer to this critical research question is in the affirmative. Specifically, with simple random instance selection followed by FR, the amount of data needed for training a classifier can be drastically reduced with minimal impact on classification performance. We also provide recommendations on which FS/ FE method to use in conjunction with which classifier.

1. Introduction

Ever-advancing technologies of information transfer and storage have enabled governments, businesses, and the research community collect data on massive scales in terms of both features and instances. In fact, it can be said that being able to record and store such huge amounts of data has become both a blessing and a curse. It is precisely for this reason that identifying a small, yet representative subset of data across features and instances has become an integral part of modern predictive analytics and pattern recognition. Excluding irrelevant and redundant features as well as removing noisy and excessive instances have numerous and rather significant advantages. These advantages include savings in storage costs, reduced computational resources required for model training and deployment, faster model training, less likelihood of overfitting, and more interpretable models. This process of data reduction across features and instances is also of utmost importance within the relatively recent paradigm of explainable artificial intelligence (XAI) that refers to AI techniques that can be easily understood and therefore trusted by humans, which is particularly relevant within business and social contexts (Dosilovic, Brcic, & Hlupic, 2018).

Two of the most effective methods for reducing feature dimensionality are feature extraction (FE) and feature selection (FS) methods. In FE, original set of features is mapped into a new set of features with a lower dimensionality. Popular FE techniques in the literature are Principle Component Analysis (PCA) (Pierce, Hope, Johnson, Wright, & Synovec, 2005), Linear Discriminant Analysis (LDA) (Fisher, 1936), Independent Component Analysis (ICA) (Stone, 2004), and ISO-Container Projection (ISOCP) (Zheng, Chenmao, & Jia, 2010). A major downside of FE methods, on the other hand, is that subsequent to feature mapping, the newly-generated features usually do not have any real-world meanings.

In comparison, FS techniques call for selecting a small subset of the original set of features by dropping irrelevant and redundant ones, so they preserve the true meanings of the original features and therefore

---

* Corresponding author.
E-mail addresses: milad@bilkent.edu.tr (M. Malekipirbazari), vural.aksakalli@rmit.edu.au (V. Aksakalli), andy.eberhard@rmit.edu.au (A. Eberhard).

https://doi.org/10.1016/j.eswa.2021.115072
Received 10 February 2020; Received in revised form 10 December 2020; Accepted 15 April 2021
Available online 23 April 2021

offer better interpretability (Aksakalli & Malekipirbazari, 2016). Feature selection methods are broadly classified into four groups: wrapper, filter, embedded, and hybrid methods. A wrapper FS method employs a particular performance criterion coupled with a specific classifier for evaluating a feature subset. On the other hand, filter FS methods work independently of any particular classifier and they measure a specific property of the data such as correlation, distance, or information gain with which features can be ranked by and consequently insignificant features can be eliminated. The reader is referred to Jović, Brkić, and Bogunović (2015) for more details on these feature selection approaches. Popular filter methods include distance-based models including ReliefF (Sikonia & Kononenko, 2003), information theory based methods such as mutual information and minimum Redundancy Maximum Relevance Feature Selection (mRMR) (Senawi, Wei, & Billings, 2017), and statistical methods including chi-squared and F-score based methods. The FE and FS methods together shall be referred to as feature reduction (FR) methods in the rest of this manuscript.

On the other hand, instance selection is another data reduction technique that can be used in many machine learning tasks. There exist several algorithms for instance selection for decreasing the original dataset to a manageable volume (Arnaiz-González, Díez-Pastor, Rodríguez, & García-Osorio, 2016; Song, Liang, Lu, & Zhao, 2017; Wilson & Martinez, 2000). An instance selection algorithm should define a subset of the total available data to accomplish the original objective of the machine learning task as if all the data had been utilized (Liu & Motoda, 2002). However, a trade-off between the dimension reduction and the classification accuracy should be dealt with by each instance selection technique. A review of instance selection methods can be found in Olvera-López, Carrasco-Ochoa, Martínez-Trinidad, and Kittler (2010).

Despite their benefits, FR methods can be a rather costly proposition in terms of execution time, especially when dealing with high dimensional data. Therefore, applying traditional FR techniques is not always a viable option in practice in big data problems (Rong, Gong, & Gao, 2019). In an attempt to address the computational challenges of FR methods for large datasets, this study proposes a simple and straightforward framework in which we only use a small random subset of instances in the dataset instead of the whole set of instances (WSI) prior to deploying an FR method. The specific research question we consider is whether random instance selection (IS) followed by FR can yield comparable (cross-validated) classification performance when compared against performance without any IS or FR as well as just FR without any IS. To our knowledge, our work is the first of its kind in the literature that investigates performance and feasibility aspects of random IS for improving performance of FR methods. We specifically aim to study relative benefits and potential downsides of this framework via comprehensive computational experiments involving 20 popular public datasets, 6 distinct classifiers, and 6 of the most common FR methods in pattern recognition. We also aim to provide simple guidelines on which FR method to use with which classifier in the process.

The datasets we consider are from a wide range of domains with number of instances ranging from 2000 to 581012 together with number of features ranging from 26 to 65770 (subsequent to one-hot-encoding of categorical features, if any). The classifiers we employ are decision trees, nearest neighbors, Naïve Bayes, support vector machines, random forests, and gradient boosting. As for FR methods, we consider the FE methods of principal components analysis and independent components analysis, and we consider the FS methods of F-score, mutual information, random forest importance, and ReliefF. Regarding number of randomly selected instances for each FR method/ classifier/ dataset combination, we consider 100, 250, 500, 1000, 1500, and 2000 instances respectively. On the other hand, in order not to over-complicate our analysis, we restrict our focus to the top 10 features as selected by each FR method. The primary statistical analysis tool we use in this study is the non-parametric Wilcoxon signed-rank test.

The rest of this manuscript is organized as follows. Section 2 presents an overview of our evaluation methodology. Section 3 describes the datasets, the FR methods, and the classifiers. Section 4 presents our computational experiments and a detailed analysis of our results. Section 5 discusses our summary and conclusions together with several directions for future research.

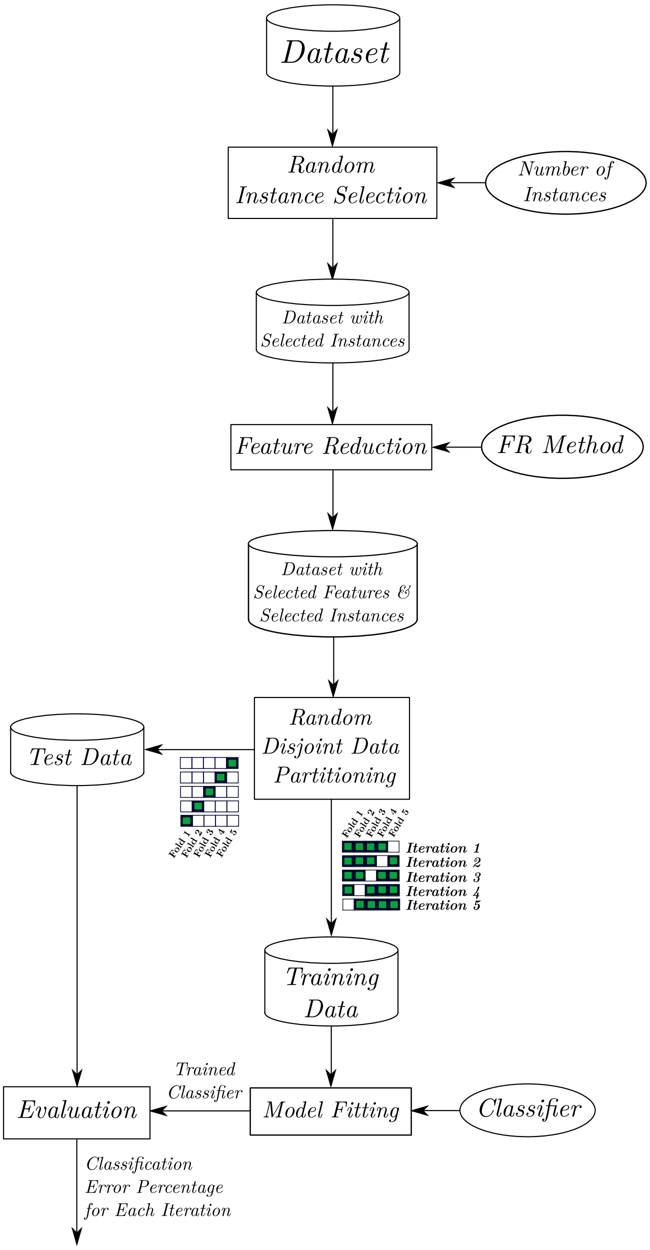


Fig. 1. Flow chart of our evaluation methodology.

2. Evaluation methodology

Our evaluation methodology is illustrated as a flow chart in Fig. 1 and it can be outlined as follows:

1. Get evaluation specifics: (i) the dataset, (ii) number of random instances to be selected, (iii) the feature reduction (feature selection or extraction) method, and (iv) the classifier.
2. Import the dataset under consideration and perform the following pre-processing operations: (i) one-hot encoding of categorical features if any, and (ii) normalization of all features between 0 and 1.
3. Randomly select a subset of the whole set of instances in the dataset for the given number of instances.

Table 1
Description of benchmark datasets. The number of features in the table are after one hot potential encoding of any categorical descriptive feature.

	Name	Instances	Features	Classes	Domain	Source	Ref.
1	Airlines	539383	608	2	Transportation	OpenML	(Vanschoren et al., 2013)
2	Bank Marketing	45211	48	2	Business	UCI	(Moro et al., 2014)
3	Connect-4	67557	42	3	Game	UCI	(Lichman, 2013)
4	Covertype	581012	54	7	Life	UCI	(Lichman, 2013)
5	Default Credit Card Clients	30000	28	2	Business	UCI	(Yeh & Lien, 2009)
6	Gisette	7000	5000	2	Computer	UCI	(Guyon et al., 2005)
7	Jannis	83733	54	4		OpenML	(Vanschoren et al., 2013)
8	Madelon	2600	500	2	Artificial	UCI	(Guyon et al., 2005)
9	Mfeat-fac	2000	216	10	Computer	UCI	(Lichman, 2013)
10	MiniBooNE	130064	50	2	Physical	UCI	(Lichman, 2013)
11	Online News Popularity	39644	59	2	Business	UCI	(Fernandes et al., 2015)
12	Phishing Websites	11055	30	2	Computer Security	UCI	(Lichman, 2013)
13	Pole-Telecommunication	14998	26	11	Telecommunication	KEEL	(Alcalá-Fdez et al., 2011)
14	Sensorless Drive Diagnosis	58509	48	11	Computer	UCI	(Lichman, 2013)
15	Spambase	4601	57	2	Computer	UCI	(Lichman, 2013)
16	Telecom Customer Churn	7043	6570	2	Business	IBM	
17	US Census Income	45222	41	2	Social	UCI	(Lichman, 2013)
18	USPS	9298	256	10	Image	ASU	(ASU, 2015)
19	Volkert	58310	180	10		OpenML	(Vanschoren et al., 2013)
20	Waveform Database Generator	5000	40	3	Physical	UCI	(Lichman, 2013)

4. For this subset of instances, use the designated feature reduction method to select the top 10 features.
5. Reduce the dataset by keeping only the features selected or extracted by the feature reduction in Step (4).
6. Apply 5-fold cross-validation to split the reduced dataset (after both instance selection and feature reduction) into training and test datasets.
7. For each of the folds, pass on the training dataset to the designated classifier for model fitting.
8. Evaluate the fitted model using the test dataset.
9. Repeat the cross-validation process 5 times and return average classification error percentage.

Within our evaluation framework, we use exactly the same random instances and also the same data partitioning for 5-repeated 5-fold cross-validation for each dataset/ number of instances/ FR method/ classifier combination. This approach ensures pairing between the combinations and reduces statistical variance, thereby increasing the accuracy of our statistical performance comparisons. Furthermore, while partitioning the dataset during each cross-validation repetition, we employ stratified partitioning (using the class labels of the target feature) for a more robust cross-validation procedure.

3. Experimental setup

This section provides details on our experimental setup including the datasets, the FR methods, and the classifiers. Our experiments were implemented in Python 3.7 using the Scikit-Learn machine learning module (Pedregosa et al., 2011) Version 0.22, and they were executed on a workstation with a 4.2 GHz octa-core CPU with 32 gigabytes of RAM.

3.1. Datasets

The 20 public datasets used in this work are described in Table 1. These datasets were retrieved from different repositories and they come from various application domains including business, telecommunications, social sciences, and computer security.[1]

3.2. Classifiers

In our experiments, we consider six different classifiers and we use the default parameter values of these classifiers in their respective implementation within the Scikit-Learn module. These classifiers were specifically chosen to be representatives of the following six major categories of supervised machine learning (Kelleher, Mac Namee, & D'Arcy, 2015): information-based learning (decision trees), similarity-based learning (nearest neighbors), error-based learning (support vector machines), probability-based learning (Naïve Bayes), bagging-based learning (random forests), and boosting-based learning (gradient boosting). We now provide a brief description of these classifiers and explain the particular parameter values used in our experiments.

- *Decision Trees (DT)*: A popular classification method that utilizes a tree-based structure with class-conditional probabilities at the tree branches. The decision tree starts with a root node and gradually builds sub-trees with internal nodes that are connected by emanating branches and ends with terminal nodes called leaves. Each internal node corresponds to a test of a feature and branches represent a binary partition of the test attribute (Aksakalli & Malekipirbazari, 2016). The decision tree implementation we use employs Gini index for splitting.
- *k-Nearest Neighbors (KNN)*: A non-parametric classification technique in which a new instance is classified based on class labels of the majority of its k nearest neighbors. Our KNN implementation uses $k = 5$ with Euclidean distance between instances.
- *Support Vector Machines (SVM)*: A supervised learning technique that maps instances such that different classes are separated by hyperplanes. For this, SVM constructs hyperplanes in a high-dimensional space and performs classification for new instances in accordance with the side of the hyperplane on which they fall (Bolón-Canedo, Sánchez-Maroño, & Alonso-Betanzos, 2015). In this work, we use $\ell 2$-penalized linear SVM with an error penalty of $C = 0.1$.
- *Naïve Bayes (NB)*: A simple probabilistic classification technique based on Bayes' theorem with an independence assumption between features when conditioned upon each class label. In this study, we use the Gaussian Naïve Bayes variant wherein features are assumed to follow a Gaussian (normal) distribution.
- *Random Forests (RF)*: A bagging-based ensemble learning technique that constructs a forest of random decision trees. RFs build multiple decision trees on bootstrapped training samples and they decorrelate the decision trees in the forest via randomization of split attributes, thereby reducing the variance when averaged over the

---

[1] These datasets are readily available on GitHub in CSV format athttps://github.com/vaksakalli/datasets.

Table 2
Description of feature reduction methods

Feature Reduction Method	FS	FE	Criterion
F-score	✓		Univariate, statistical
Mutual Information (MutualInfo)	✓		Univariate, information-theoretic
ReliefF	✓		Multivariate, distance-based
Random Forest Importance (RFI)	✓		Multivariate, computed from permuting out-of-bag data
Principal Component Analysis (PCA)		✓	Multivariate, unsupervised
Independent Component Analysis (ICA)		✓	Multivariate, unsupervised

trees (Breiman, 2001). In our implementation, we use 100 decision trees for each RF model.

- *Gradient Boosting (GB)*: A boosting-based ensemble learning technique that can be seen as an iterative functional gradient descent algorithm. Similar to other boosting methods, GB combines weak learners, which are a total of 100 decision trees in our implementation, into a single strong learner in an iterative fashion (Friedman, 2002).

3.3. Feature selection & feature extraction methods

In our experiments, we employ a total of six popular feature reduction (FR) techniques, four of which are feature selection (FS) and two are feature extraction (FE) methods. With respect to FS methods, we first run the method to rank the features with respect to their importance, and then we pick the top 10 features. All the FS methods we consider are filter methods, so the top 10 features selected are independent of the intended classifier. Regarding FE methods, we first map the original feature space into a lower dimensional space using the particular FE method and then choose the top 10 features in this lower dimensional space. As in classifiers, we used the default parameter values of the FR methods in their respective implementation within Scikit-Learn. A summary of these methods is given in Table 2 and the methods are briefly described below.

- *F-score*: A statistical FS method that measures the discriminative power of each descriptive feature (with respective to the target feature) independently from other descriptive features. Once F-score of each descriptive feature is calculated, these features are ranked based on their F-scores, with higher values indicating higher importance (Polat & Günes, 2009).
- *Mutual Information*: An information-theoretic FS method that measures the dependency between each descriptive feature and the target feature as the mutual information between them, with higher values indicating higher feature importance. The particular implementation we employ utilizes a non-parametric method based on entropy estimation from k-nearest neighbors distances (Cover, 2006).
- *ReliefF*: An extension of the original Relief algorithm that accounts for multivariate feature interactions. ReliefF assigns scores to each feature and updates the scores in each iteration until termination. It randomly selects an instance from the data and increases the feature score if a feature value difference is detected in a neighboring instance pair with the same class label, and it decreases the feature score otherwise (Bolón-Canedo et al., 2015).
- *Random Forest Importance (RFI)*: The random forest classifier used as a filter FS method. In RFI, first a random forest classifier is fit to the training data. Next, for each tree in the forest, out-of-bag instances for that tree are randomly permuted for each feature and decrease in performance is recorded. This decrease is then averaged across all the trees. Importance of a feature is determined by the amount of decrease in performance across the ensemble of the fitted trees (Everson et al., 2015).
- *Principal Component Analysis (PCA)*: An unsupervised FE method for reducing dimensionality of multivariate data. PCA projects the original data onto a new set of orthogonal features (principal components) using the data covariance matrix. In PCA, each data point in the original space can be expressed as a linear combination of these principal components, which are the new features. Importance of these new features are determined by the amount of variance captured by each feature (Pierce et al., 2005).
- *Independent Component Analysis (ICA)*: An unsupervised FE method for separating multivariate data into additive subcomponents that are statistically independent from each other. Whereas the new features found by PCA are orthogonal, this is not necessarily the case in ICA. In particular, the goal in PCA is to compress data whereas the goal in ICA is to separate data (Stone, 2004).

3.4. Performance criterion

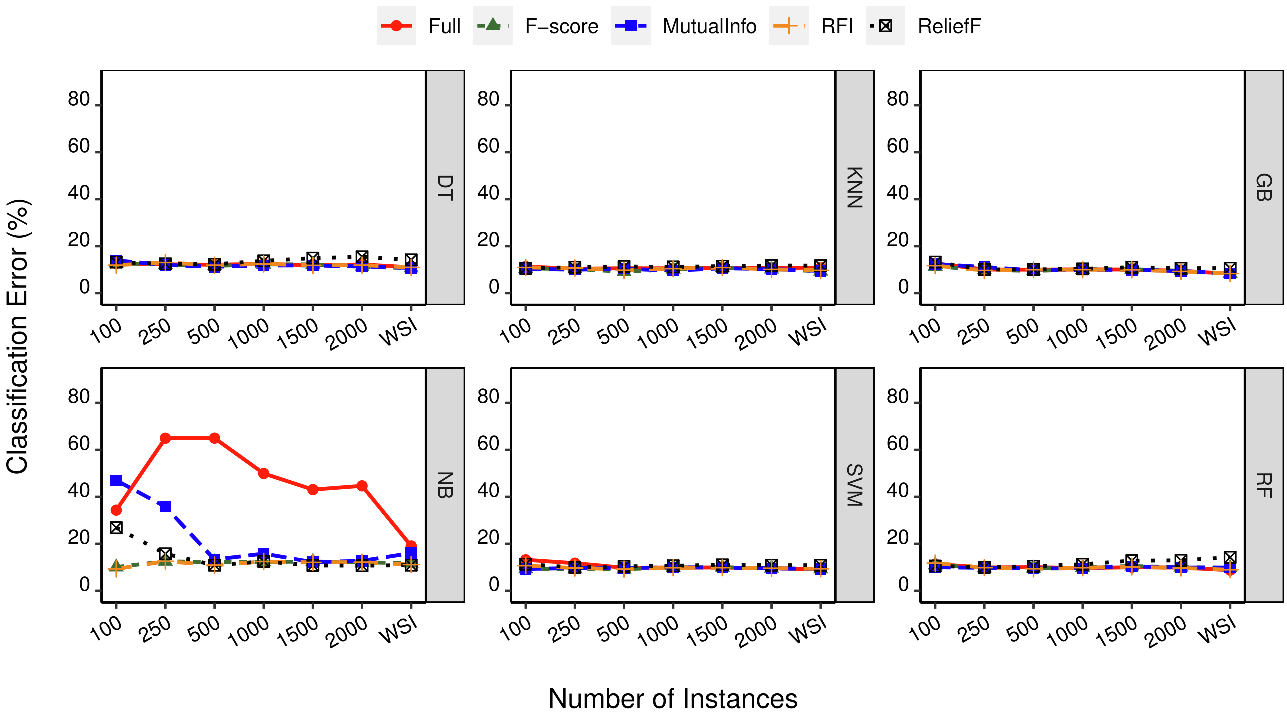
The performance criterion in our experiments is taken as the 5-repeated 5-fold cross-validation error percentage. In 5-fold cross-validation, the original data is first split into five equal-sized partitions. For each one of the 5 partitions, this particular partition is set aside for testing and the rest are used for training, which results in 5 classification error percentages in total, which are then averaged. Cross-validation error is a well-known and popular performance evaluation metric that also mitigates the overfitting issue of conventional evaluation strategies and ensures generalized performance for classifiers beyond the available data (Wong, 2015). In addition, repeating the cross-validation process several times as in our study helps reduce potential biases that might be introduced during the data partitioning step, resulting in a more robust evaluation process.

4. Computational experiments

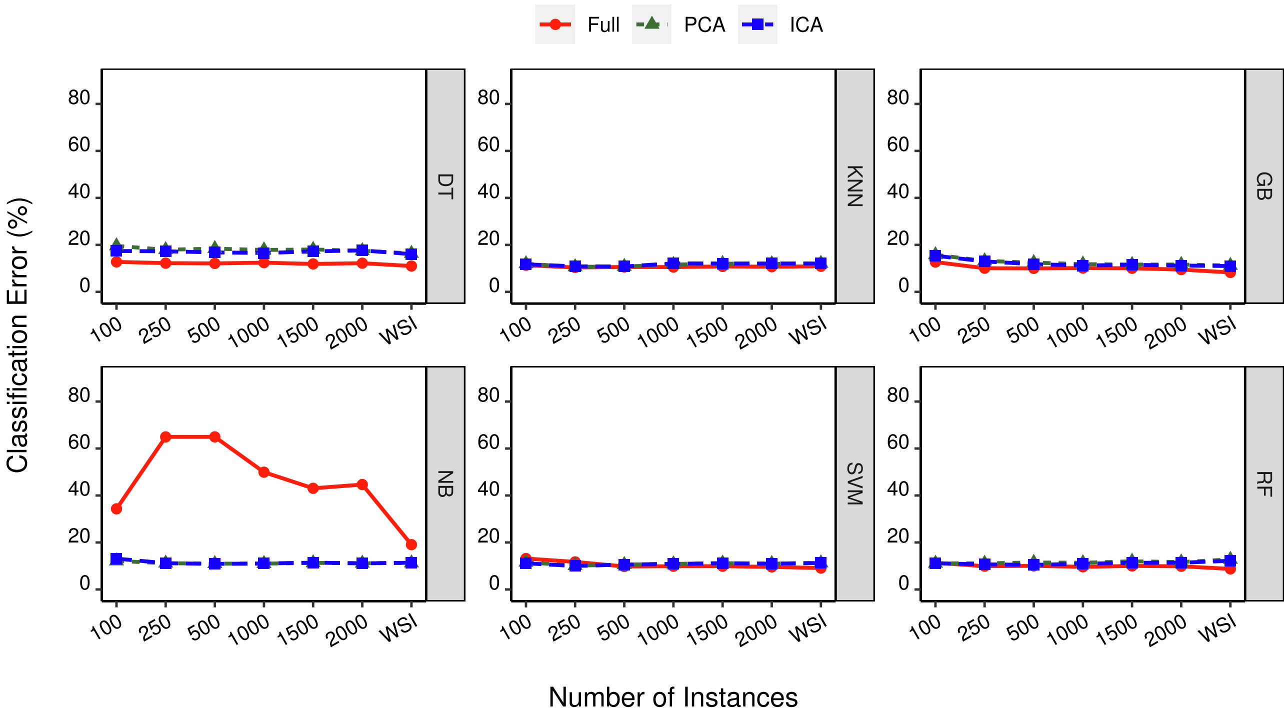
As mentioned earlier, our computational experiments were carried out in a paired fashion in the sense that for a given dataset and a given number of instances, all FR methods were executed on the same subset of random instances. Furthermore, subsequent to FR, all classifiers were fitted on the same cross-validation (CV) train partitions and all evaluations were done on the same CV test partitions. Once the CV error percentages were calculated, we utilized the Wilcoxon signed-rank test (Wilcoxon, 1945) on these percentages at a 5% significance level in order to detect any statistically significant performance differences.

Figs. 2–5 show mean CV error percentages for benchmark datasets from the business domain for each different number of randomly selected instances and for each FR method and classifier combination. In these figures, Parts (a) show the comparison results for FS methods whereas Parts (b) show the comparison results for FE methods. As can be seen in these figures, the Naïve Bayes classifier appears to exhibit a considerable variation in performance for a given dataset across our experiments. This can perhaps be attributed to the fact that Naïve Bayes is quite sensitive to class priors and each time a random instance selection is performed, the prior probabilities change, resulting in relatively large variation in classification performance.

Table 3 presents percentage point increase in average CV errors across all datasets for number of selected instances ranging from 100 to 2000 compared to the whole set of instances (WSI). As an example, with F-score as the FR method and DT as the classifier with 100 randomly selected instances, CV error increases by 7.3 percentage points (with a standard error of 0.84 percentage points) compared to the case with WSI (i.e, without any instance selection). In the table, “Full” denotes the full set of features.

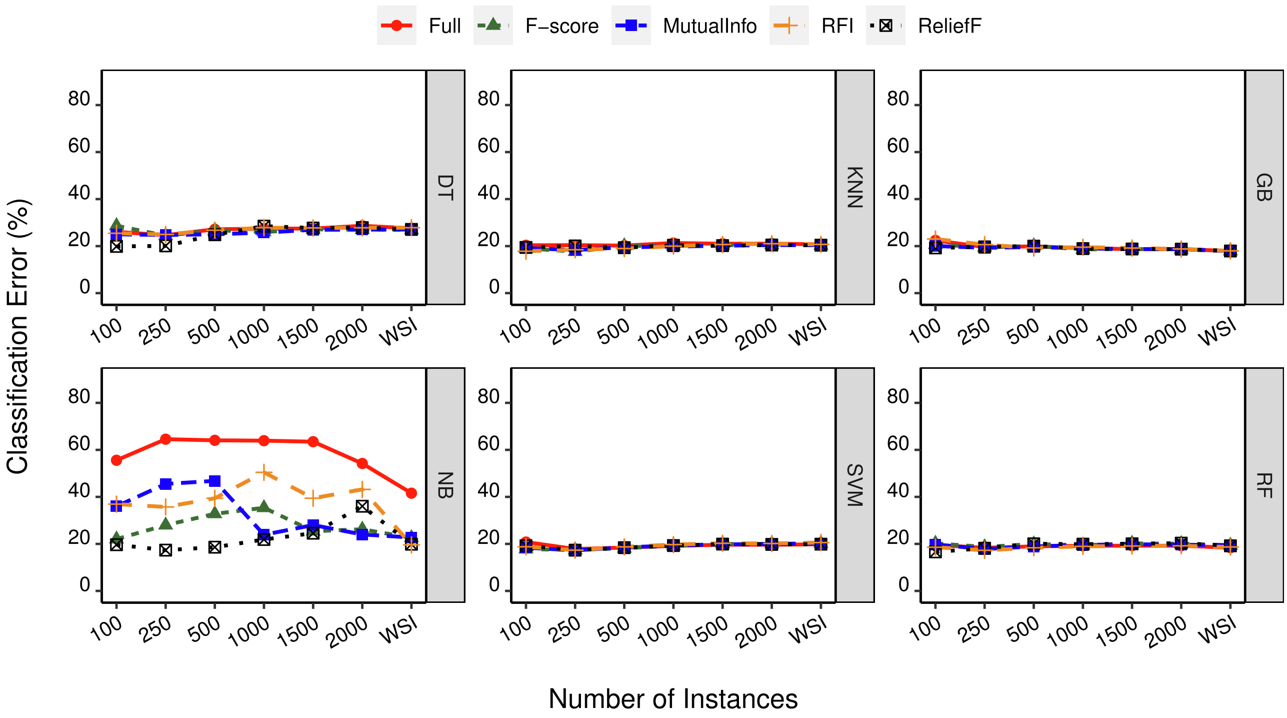


(a) Comparison between feature selection methods

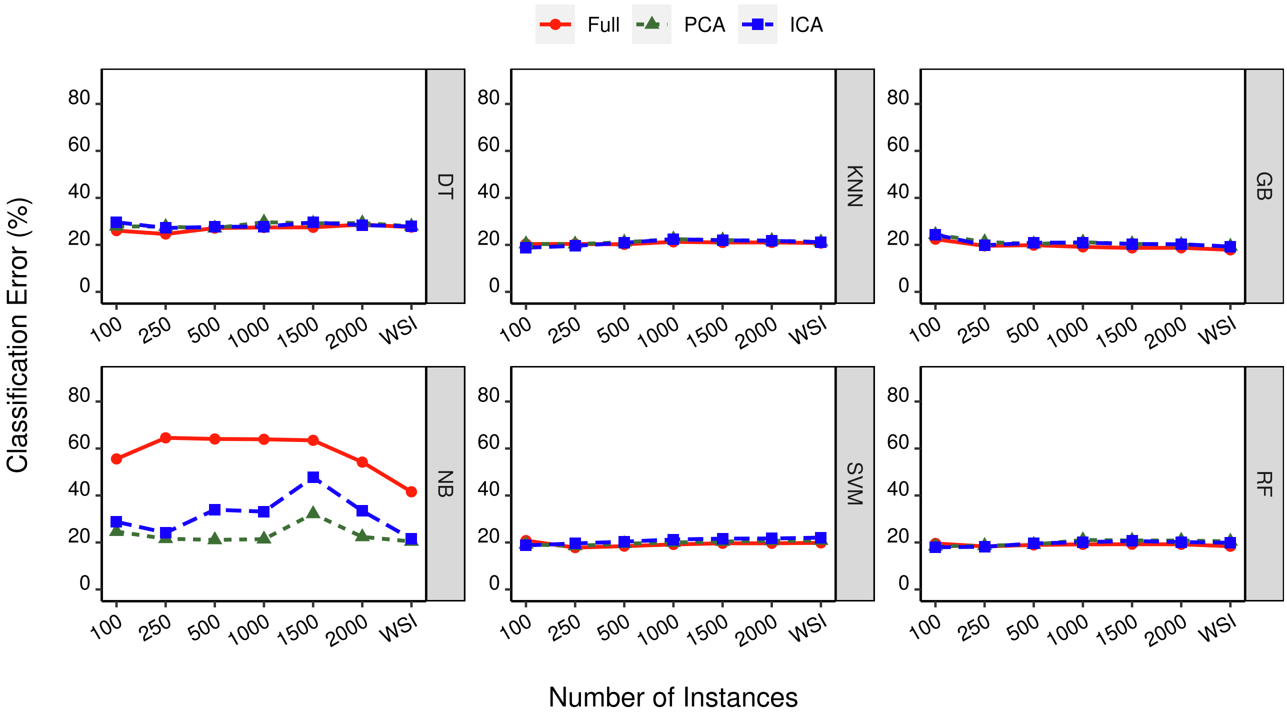


(b) Comparison between feature extraction methods

Fig. 2. Mean CV error percentages for different classifiers and number of randomly selected instances for “Bank Marketing” dataset (with size 42118 × 62 and 2 different classes).

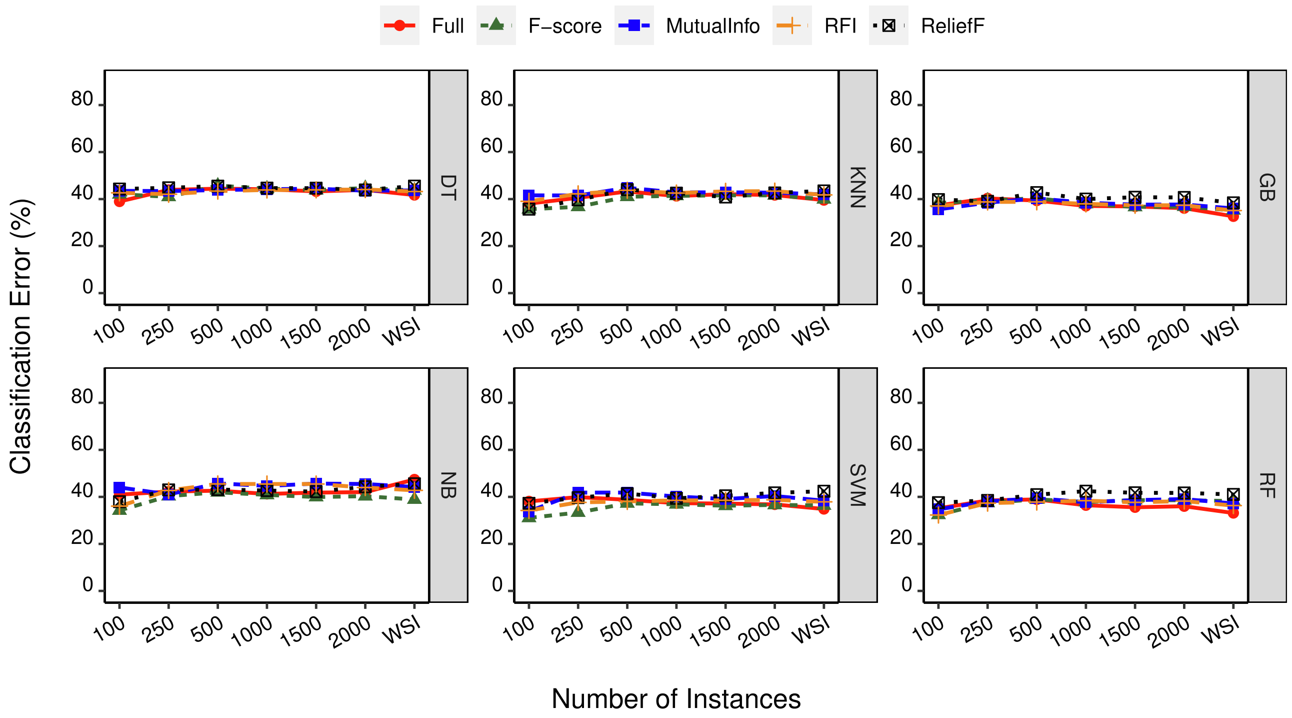


(a) Comparison between feature selection methods

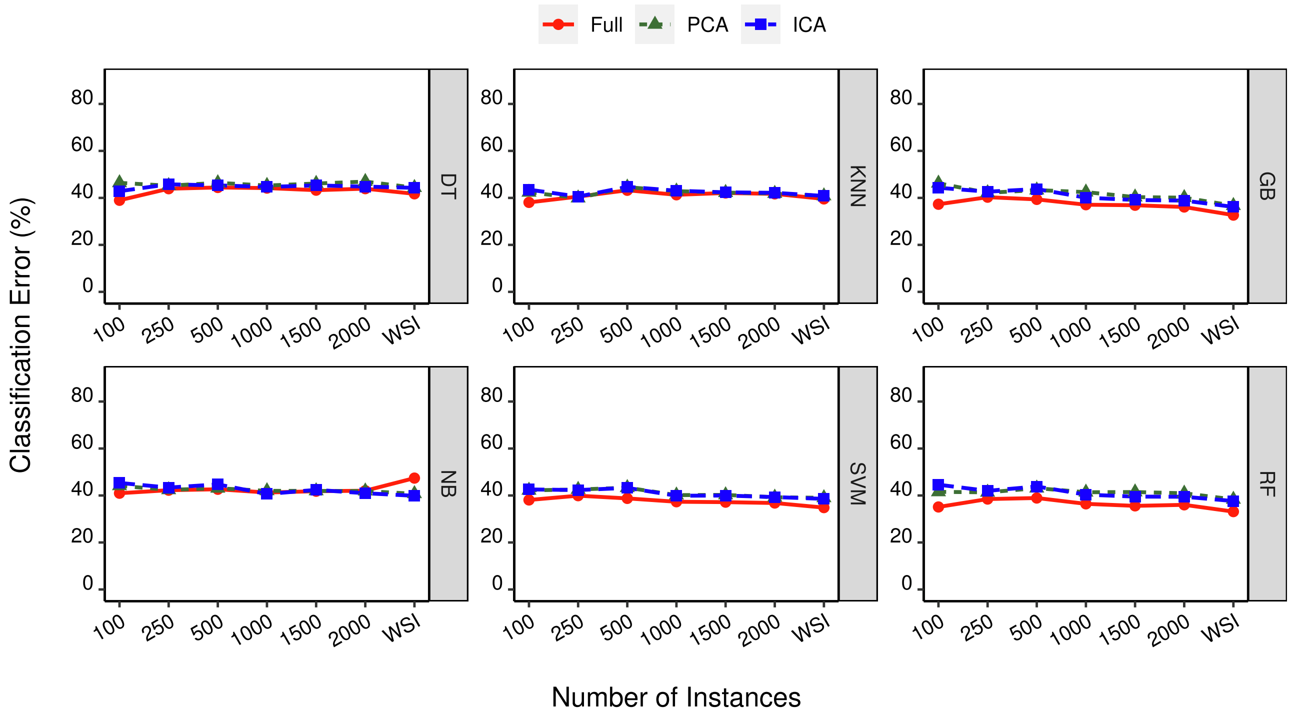


(b) Comparison between feature extraction methods

Fig. 3. Mean CV error percentages for different classifiers and number of randomly selected instances for “Default Credit Card Clients” dataset (with size 30000 × 28 and 2 different classes).

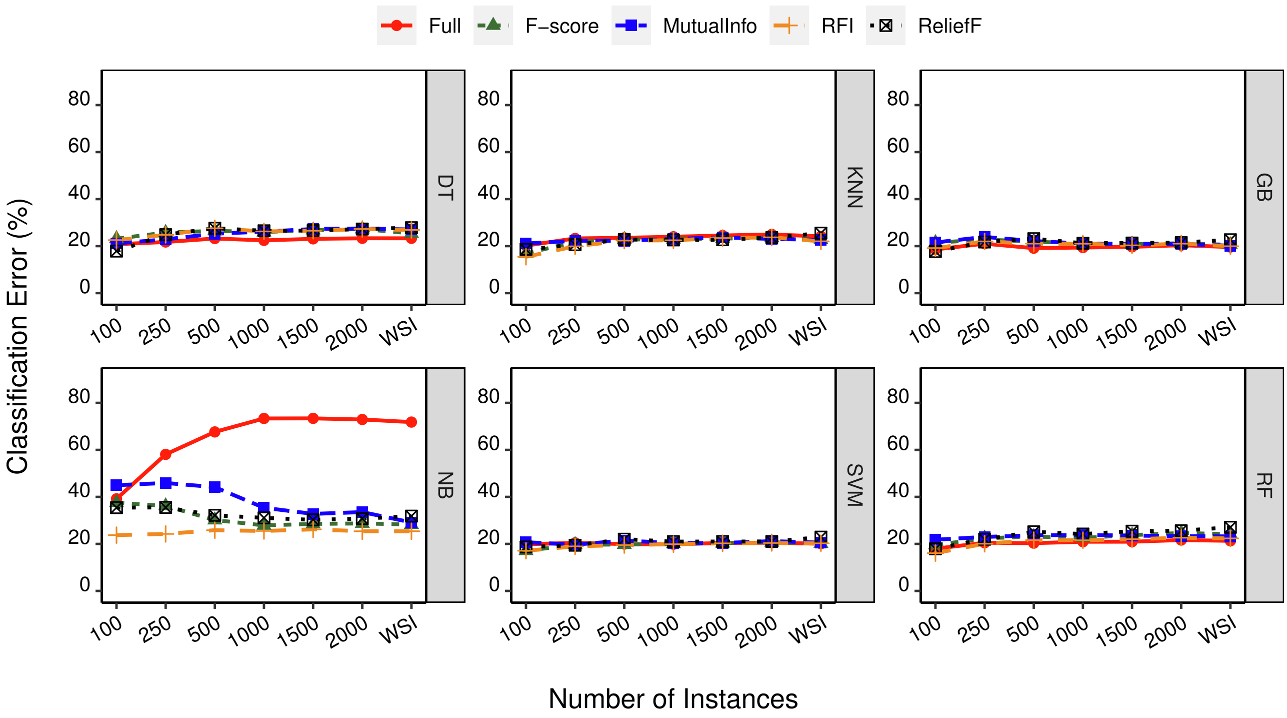


(a) Comparison between feature selection methods

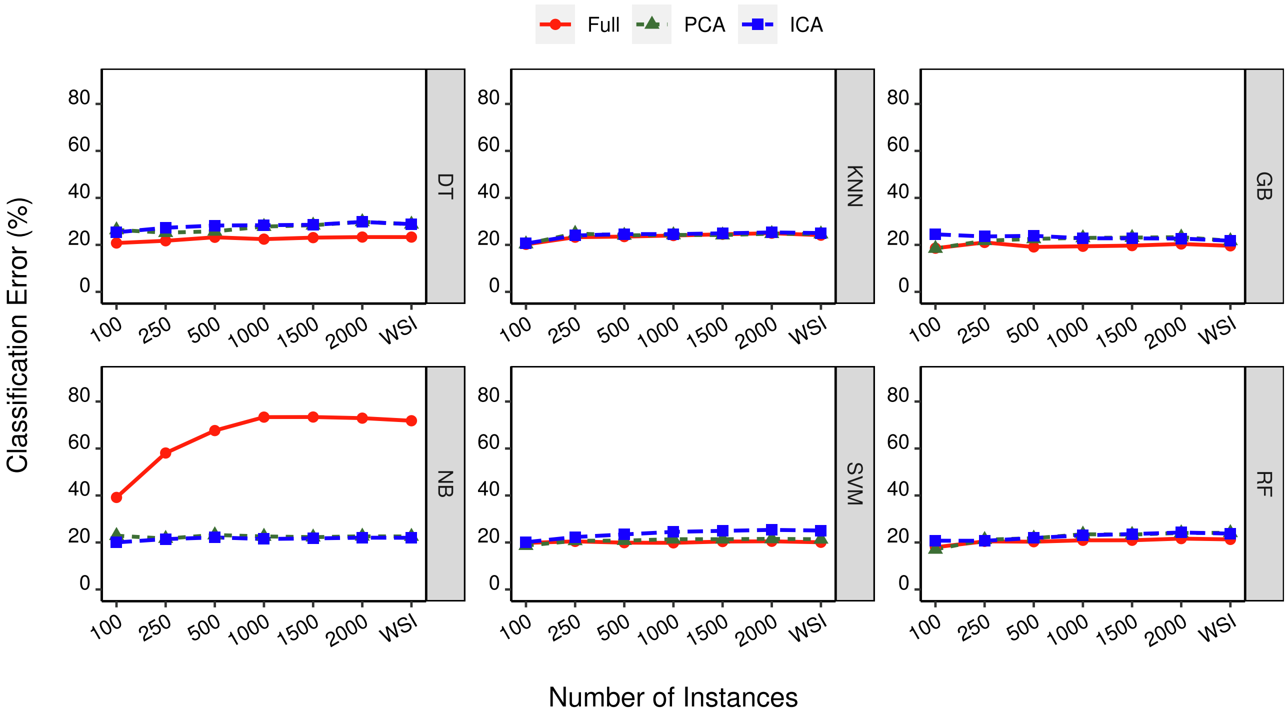


(b) Comparison between feature extraction methods

Fig. 4. Mean CV error percentages for different classifiers and number of randomly selected instances for “Online News Popularity” dataset (with size 39644 × 59 and 2 different classes).



(a) Comparison between feature selection methods



(b) Comparison between feature extraction methods

Fig. 5. Mean CV error percentages for different classifiers and number of randomly selected instances for “Telco Customer Churn” dataset (with size 7043 × 6570 and 2 different classes).

Table 3
Percentage point increase in average CV errors across all datasets for number of selected instances ranging from 100 to 2000 compared to WSI. Standard errors are shown with a ± sign. The best result for each FR method/ classifier/ number of instances combination is shown in bold.

	Feature Reduction Method	Number of Instances					
		100	250	500	1000	1500	2000
DT	F-score	7.3 ± 0.84	6.07 ± 0.69	5.9 ± 0.57	4.03 ± 0.44	3 ± 0.34	**2.43** ± 0.3
	MutualInfo	7.09 ± 0.89	6.04 ± 0.72	5.07 ± 0.62	4.03 ± 0.54	3.79 ± 0.44	**2.91** ± 0.45
	ReliefF	9.78 ± 2.32	5.36 ± 1.02	3.99 ± 0.73	3.13 ± 0.59	2.3 ± 0.55	**1.47** ± 0.47
	RFI	8.28 ± 0.9	7.3 ± 0.77	5.71 ± 0.61	4.61 ± 0.51	4.09 ± 0.49	**3.39** ± 0.46
	PCA	13.6 ± 1.53	10.07 ± 1.13	7.73 ± 0.91	6.46 ± 0.76	5.13 ± 0.58	**4.57** ± 0.54
	ICA	15.47 ± 1.66	11.78 ± 1.24	9.14 ± 0.99	6.79 ± 0.75	5.54 ± 0.61	**4.73** ± 0.58
	Full	12.6 ± 1.07	9.07 ± 0.78	7.68 ± 0.65	5.92 ± 0.55	4.96 ± 0.5	**4.27** ± 0.46
GB	F-score	8.9 ± 0.79	6.72 ± 0.57	5.39 ± 0.48	3.34 ± 0.39	2.13 ± 0.28	**1.56** ± 0.2
	MutualInfo	8.49 ± 0.82	6.19 ± 0.48	5.15 ± 0.43	3.36 ± 0.27	2.63 ± 0.25	**2.26** ± 0.24
	ReliefF	11.21 ± 2.19	5.95 ± 0.89	4.29 ± 0.67	2.69 ± 0.51	1.93 ± 0.46	**1.29** ± 0.39
	RFI	9.26 ± 0.86	6.76 ± 0.56	4.95 ± 0.46	3.41 ± 0.3	2.85 ± 0.28	**2.15** ± 0.21
	PCA	14.18 ± 1.5	9.2 ± 0.97	6.49 ± 0.67	4.36 ± 0.46	3.11 ± 0.34	**2.48** ± 0.27
	ICA	15.6 ± 1.39	9.88 ± 0.96	7.16 ± 0.73	4.36 ± 0.47	3.04 ± 0.33	**2.49** ± 0.3
	Full	13.61 ± 1.12	8.65 ± 0.67	5.86 ± 0.48	3.92 ± 0.32	2.9 ± 0.27	**2.32** ± 0.23
KNN	F-score	5.54 ± 0.97	5.18 ± 0.76	4.71 ± 0.66	3.15 ± 0.43	2.17 ± 0.3	**1.63** ± 0.24
	MutualInfo	6.62 ± 1.07	6.37 ± 0.93	5.27 ± 0.62	3.64 ± 0.45	3.06 ± 0.41	**2.36** ± 0.38
	ReliefF	9.71 ± 2.67	5.46 ± 1.34	3.51 ± 0.9	2.14 ± 0.73	1.1 ± 0.66	**0.9** ± 0.59
	RFI	7.99 ± 1.18	6.33 ± 0.88	5.41 ± 0.66	4.06 ± 0.51	3.32 ± 0.46	**2.98** ± 0.42
	PCA	12.8 ± 1.66	9.24 ± 1.35	7.58 ± 1.01	5.7 ± 0.71	4.54 ± 0.6	**3.76** ± 0.5
	ICA	12.36 ± 1.38	8.65 ± 1.07	7.54 ± 0.86	5.49 ± 0.63	4.6 ± 0.57	**3.95** ± 0.52
	Full	12.08 ± 1.69	9.44 ± 1.32	7.65 ± 1.08	6.09 ± 0.84	5.27 ± 0.71	**4.58** ± 0.62
NB	F-score	−**1.93** ± 1.84	0.21 ± 1.77	2.32 ± 1.35	2.92 ± 1.21	0.68 ± 0.81	0.51 ± 0.67
	MutualInfo	2.57 ± 2.15	4.52 ± 2.14	5.24 ± 1.61	3.06 ± 1.24	3.54 ± 1.28	**2.38** ± 1.03
	ReliefF	6.06 ± 2.41	3.93 ± 1.64	2.84 ± 1.46	0.83 ± 0.52	**0.24** ± 0.44	1.13 ± 0.87
	RFI	**0.36** ± 1.7	0.03 ± 1.46	1.42 ± 1.01	1.44 ± 0.98	0.43 ± 0.78	0.72 ± 0.85
	PCA	6.27 ± 1.06	2.39 ± 0.71	1.77 ± 0.54	1.22 ± 0.44	1.26 ± 0.51	**0.8** ± 0.33
	ICA	5.97 ± 1.21	2.64 ± 0.83	3.1 ± 0.85	2.23 ± 0.76	2.53 ± 0.92	**2.03** ± 0.7
	Full	−**2.71** ± 2.01	2.98 ± 2.09	5.13 ± 1.78	5.16 ± 1.6	4.68 ± 1.45	4.5 ± 1.35
RF	F-score	5.9 ± 0.8	5.31 ± 0.64	4.71 ± 0.54	3.22 ± 0.43	2.34 ± 0.28	**1.88** ± 0.24
	MutualInfo	5.39 ± 0.8	5.41 ± 0.6	4.53 ± 0.5	3.33 ± 0.46	3.04 ± 0.41	**2.54** ± 0.38
	ReliefF	7.96 ± 2.14	4.2 ± 0.98	3.17 ± 0.7	2.26 ± 0.5	1.65 ± 0.45	**1.13** ± 0.39
	RFI	6.57 ± 0.85	5.98 ± 0.68	4.63 ± 0.55	3.68 ± 0.41	3.24 ± 0.37	**2.72** ± 0.36
	PCA	11.81 ± 1.49	8.48 ± 1.01	6.6 ± 0.78	5.1 ± 0.57	4.08 ± 0.48	**3.34** ± 0.42
	ICA	13.48 ± 1.53	9.4 ± 1.04	7.47 ± 0.85	5.22 ± 0.6	4.27 ± 0.51	**3.6** ± 0.47
	Full	10.09 ± 0.92	7.32 ± 0.68	5.69 ± 0.53	4.28 ± 0.46	3.57 ± 0.42	**3.17** ± 0.4
SVM	F-score	0.81 ± 0.79	1.02 ± 0.53	1.3 ± 0.41	0.83 ± 0.33	0.44 ± 0.23	**0.3** ± 0.21
	MutualInfo	1.8 ± 0.6	1.72 ± 0.45	1.68 ± 0.33	0.84 ± 0.24	0.68 ± 0.18	**0.61** ± 0.16
	ReliefF	3.87 ± 1.5	1.34 ± 0.6	0.46 ± 0.4	−0.03 ± 0.31	−0.3 ± 0.26	−**0.44** ± 0.23
	RFI	1.33 ± 0.65	1.03 ± 0.39	0.77 ± 0.24	0.5 ± 0.21	0.37 ± 0.18	**0.12** ± 0.17
	PCA	4.71 ± 0.82	2.05 ± 0.49	1.08 ± 0.34	0.62 ± 0.22	0.39 ± 0.12	**0.2** ± 0.11
	ICA	2.44 ± 0.68	0.82 ± 0.39	0.45 ± 0.26	0.26 ± 0.22	0.03 ± 0.23	−**0.17** ± 0.19
	Full	8.29 ± 0.8	4.97 ± 0.55	3.52 ± 0.38	2.34 ± 0.27	1.7 ± 0.21	**1.24** ± 0.17

4.1. Analysis of the results

In this section, we analyze the results of our experiments from three different perspectives:

- Prediction performance of classifiers on each dataset without any instance selection or feature reduction
- Comparison of FR methods with respect to each classifier without any instance selection
- Impact of the amount of instances on the FR methods across the classifiers

4.1.1. Full set of features and instances

Here, we focus on the performance of the classifiers with full set of features and whole set of instances. The best classifier for each dataset with respect to CV errors is reported in Table 4. Also, the average execution time of these classifiers across all the datasets is displayed in Fig. 6. Unsurprisingly, the results are generally in favor of the ensemble learners in terms of classification performance. Out of the 20 datasets, ensemble methods of gradient boosting (GB) and random forests (RF) showed the best performance for 15 of the datasets. In addition, most datasets with number of instances more than 10,000 have either GB or RF as their best classifier. The strength of these models comes from the fact that they combine individual models which makes them more flexible and less sensitive to noise compared to conventional models. On

Table 4
Best performance for the benchmark datasets without any instance selection or feature reduction.

Dataset	Best Classifier	Mean Error %	NSWC†
Airlines	GB	35.51 ± 0.17	–
Bank Marketing	GB	8.27 ± 0.03	–
Connect-4	RF	18.54 ± 0.03	–
Covertype	RF	12.58 ± 0.06	–
Default Credit Card Clients	GB	17.89 ± 0.02	–
Gisette	RF	2.85 ± 0.05	GB & SVM
Jannis	RF	29.59 ± 0.02	–
Madelon	DT	25.53 ± 0.40	GB
Mfeat-fac	SVM	1.93 ± 0.04	–
MiniBooNE	RF	6.37 ± 0.01	–
Online News Popularity	GB	32.67 ± 0.06	–
Phishing Websites	RF	2.79 ± 0.03	–
Pole-Telecommunication	RF	13.77 ± 0.05	–
Sensorless Drive Diagnosis	RF	0.14 ± 0.00	–
Spambase	RF	4.66 ± 0.06	–
Telecom Customer Churn	GB	19.59 ± 0.12	–
US Census Income	GB	13.68 ± 0.02	–
USPS	KNN	3.11 ± 0.02	–
Volkert	KNN	31.42 ± 0.03	–
Waveform Database Generator	SVM	13.25 ± 0.05	–

†Not statistically worse classifiers

the other hand, GB was not superior in predicting datasets with more than two different classes, and in general it had a much longer execution time compared to other methods with the fastest classifier being Naïve Bayes.

4.1.2. *Comparison between FR methods*

In this section, we compare the performance of FR methods across all datasets in two cases of using whole set of instances and using 2000 randomly selected instances. The results of these comparisons are given in Fig. 7 wherein average CV error percentages and their standard errors are shown for each FR method/ classifier combination. In addition, statistically better methods for different classifiers with WSI are reported in Table 5 for FS and FE methods respectively. We observe the exact same pattern for the case of 2000 instances. According to this evaluation, performances of the FE method PCA and the FS method RFI are superior to that of the other FE/ FS techniques respectively. Specifically, among feature selection methods, RFI performs better with classifiers of DT, GB, and RF, all of which are tree-based classifiers. F-score, however, exhibits good performance with SVM, but performs poorly with RF. Regarding a general comparison between the best FS method of RFI vs. the best FE method of PCA, there does not seem to be large difference other than the case of NB classifier where PCA performs significantly better than RFI. Per Table 3, deploying an FR method (with 10 features) with whole set of instances results in more than a quarter percentage point *decrease* in CV error on the average across our experiments when compared against CV error without any feature reduction. The same comparison against CV errors without feature reduction but with 2000 instances shows more than a quarter percentage point *increase* in CV error.

Fig. 8 presents a comparison between average FR plus model fitting run time averaged over all 20 datasets and 6 classifiers with respect to different FR methods for 2000 randomly selected instances in contrast with WSI. In addition, Table 6 shows comparison between mean run time for FR plus CV evaluation together with CV errors averaged over all datasets and classifiers with respect to feature reduction processes for 2000 randomly selected instances in comparison with WSI. We observe in our experiments that instance selection with 2000 and 1000 instances result in about 50-fold and 65-fold reduction in run time respectively for FR methods whereas the percentage point increase in CV errors is only 1% for 2000 instances and 2% for 1000 instances.

4.1.3. *Effects of random instance selection*

As Table 3 suggests, mean CV error percentage slightly increases as the number of random instances decreases for most of the FR method/ classifier combinations when considered across all the datasets. In order to quantify this increase in CV error, we test for the following hypotheses at a 5% significance level using the Wilcoxon signed-rank test:

$$\frac{\mu_{RIS} - \mu_{WSI}}{\mu_{WSI}} \times 100 \geqslant \theta, \qquad (1)$$

where μ_{WSI} and μ_{RIS} denote mean CV error percentages of classifiers with WSI and randomly instance selection (RIS) respectively. Here, θ represents a specific percentage across the range of 0 to 20% in 1% increments.

We test each FR method/ classifier/ number of instances combination and, report the results for different number of instances over all classifiers in Fig. 9. Results for individual classifiers are shown in Fig. 10. These results indicate that classification performance with randomly selected instances of 2000 and 1000 is never statistically worse than 12% and 16% respectively, when compared to the FR performance with WSI. In regards to specific classifiers, we notice that for the NB and SVM classifiers, classification performance with randomly selected 2000 instances is never statistically worse than 2%, again when compared to the

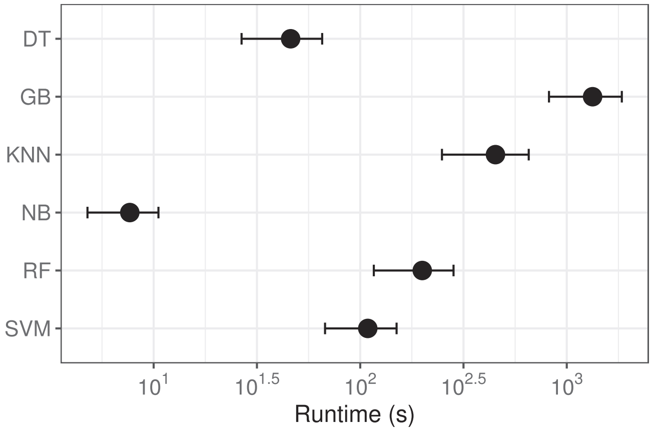
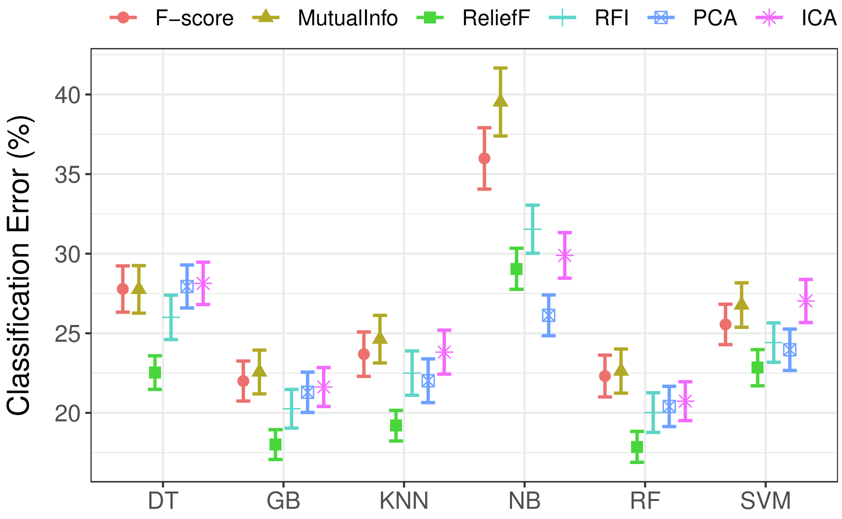
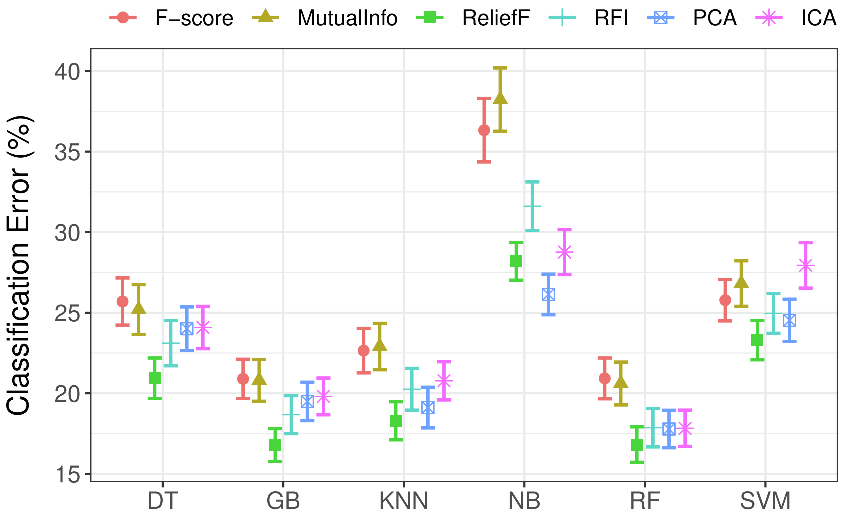


Fig. 6. Comparison between mean classification run time averaged over all datasets with full features and instances with respect to different classifiers. Standard errors are shown as error bars.



(a) WSI



(b) 2000 instances

Fig. 7. Comparison between performance of FR method/ classifier combinations with (a) whole set of instances and (b) 2000 randomly selected instances averaged over all benchmark datasets.

Table 5
Best FR method for different classifiers with WSI. The best method is shown with a circled check mark. Not statistically worse methods compared to the best method are shown with a simple check mark.

Feature Reduction Method	Classifier DT	GB	KNN	NB	RF	SVM
FS Method						
F-score						
MutualInfo						
ReliefF						
RFI	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
FE Method						
PCA	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
ICA	✓	✓				✓

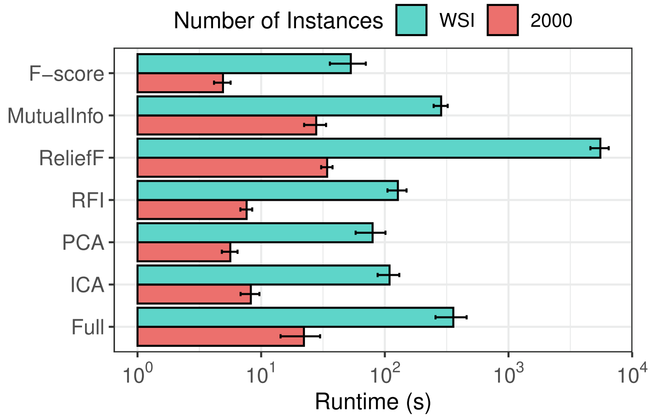


Fig. 8. Comparison between average FR plus model fitting run time averaged over all datasets and classifiers for 2000 randomly selected instances in comparison with WSI. Standard errors are shown as error bars. “Full” indicates classifier model fitting run times without any feature reduction.

Table 6
Comparison between mean FR plus model fitting run time and CV errors averaged over all datasets and classifiers with respect to FR methods for 2000 randomly selected instances in comparison with WSI.

Feature Reduction Method	Runtime with WSI (s)	Runtime for 2000 instances (s)	Runtime ratio	Mean CV error % point increase
F-score	53.1	4.9	10.8	0.31
MutualInfo	285.3	27.9	10.2	0.98
ReliefF	5530.3	34.1	161.8	0.87
RFI	127.5	7.6	16.8	0.77
PCA	79.6	5.6	14.2	1.09
ICA	109.2	8.2	13.3	1.30

FR performance with WSI.

4.1.4. Statistical variation in random instance selection

In our evaluation methodology, we perform a single run of random instance selection for a given sample size for each dataset across the cross validation process. However, in order to measure how dependent the results are on the selected subset of instances, we perform 25 repetitions of random instance selection for each experimental combination and measure the statistical variation in the random instance selection process. Fig. 11 and Table 7 show the variability of the classification error for the case of 2000 randomly selected instances for the largest five datasets. These results indicate that there is little variation in the subset of selected instances in our framework and the final performance is robust to the selection process. In particular, the standard errors (that is, standard deviations divided by the square root of the sample size of 25) across the 20 datasets has a mean and standard error of 0.689 and 0.03 respectively. To be clear, the reported value of 0.03 here is the standard error of the standard errors across the 20 datasets for the classification error percentage. In addition, the mean standard error for each classifier is less than 0.5 with the exception of Naïve Bayes which has a mean standard error of 2.13.

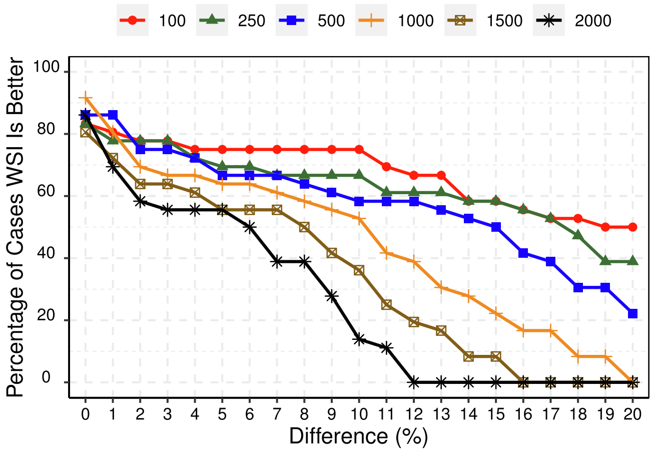


Fig. 9. Percentage of cases where percentage point increase of average CV errors when compared to WSI is statistically larger than values ranging from 0 to 20%, across all classifiers, FR methods, and datasets.

5. Summary and conclusions

In this study, we evaluate a new framework for classification with large datasets in which we first perform random instance selection (RIS) followed by feature reduction (FR) (either feature selection (FS) or feature extraction (FE)) with 10 features, which we denote by RIS + FR. This approach aims to address the issue of excessive run times of these FR methods when executed on large datasets. We present computational experiments on 20 large-scale public datasets with 6 distinct classifiers (DT, KNN, NB, SVM, GB, RF) and 6 FR methods (F-score, MutualInfo, ReliefF, RFI, PCA, ICA). In these experiments, we employ 5-repeated 5-fold cross-validation (CV) error percentage as the performance criterion of each experimental combination.

Per our computational experiments, without any IS or FR on the datasets, ensemble methods of GB and RF were able to outperform the other classifiers for most datasets, yet run time of RF was significantly shorter compared to GB. Regarding FR methods with whole set of instances (WSI), the performances of PCA as an FE method and RFI as an FS method were superior to other FE and FS methods respectively. Given the fact that, all else equal, FS methods are preferred to FE methods as they maintain the real-world meaning of the original features, our general recommendation would be to use RFI as the preferred FR method for all classifiers except for NB for which we recommend PCA due to its superior performance.

Across our experiments, without any instance selection, we observe that deploying an FR method results in more than 0.25 percentage point *reduction* in CV error on the average when compared against CV errors without FR, illustrating usefulness of FR methods in big data problems in general. In regards to RIS + FR, that is, by randomly sampling a small proportion of WSI followed by FR, our results indicate that similar classification performances can be achieved with drastically shorter FR run times. Specifically, we observe that randomly selecting 2000 and

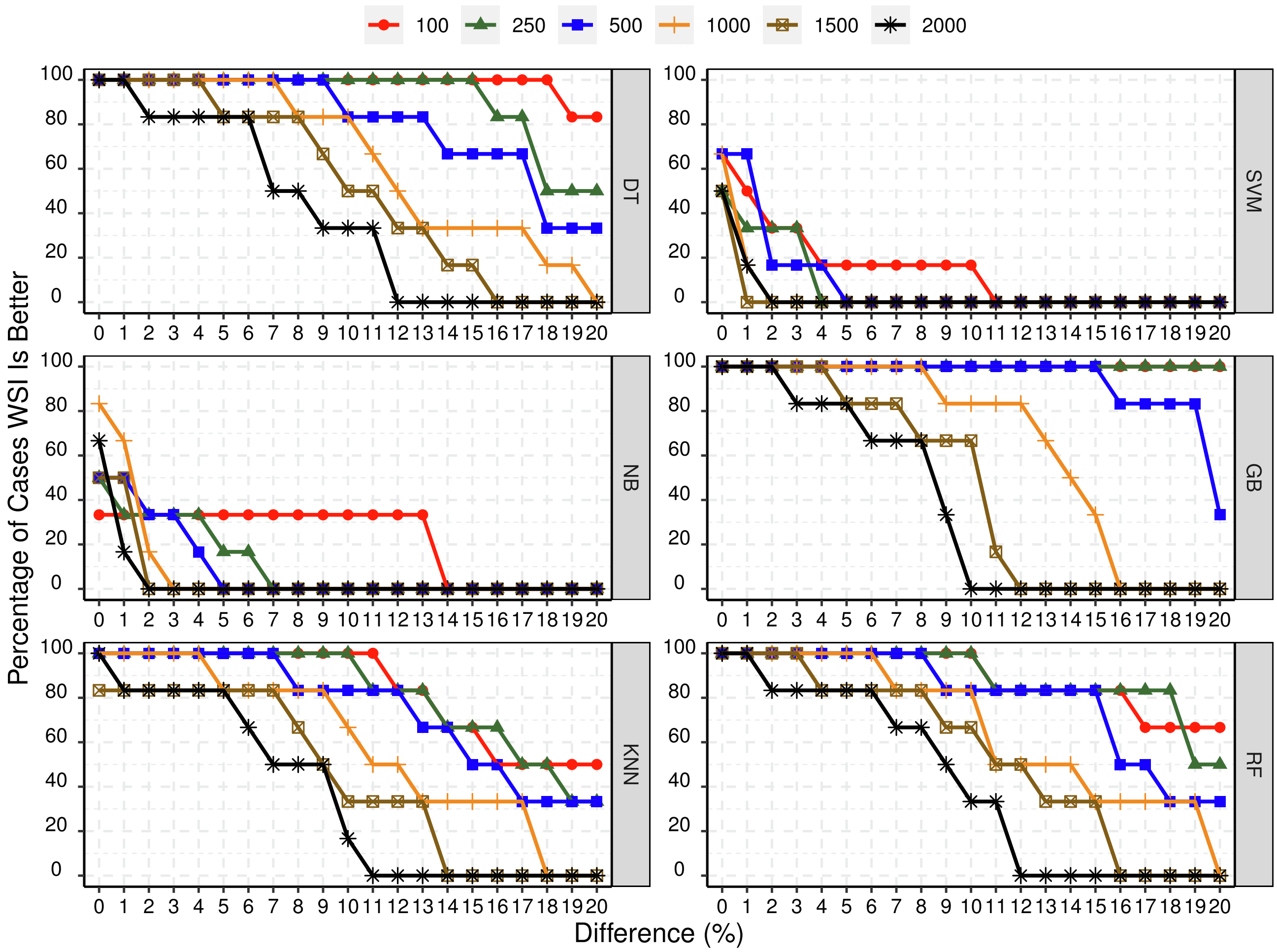
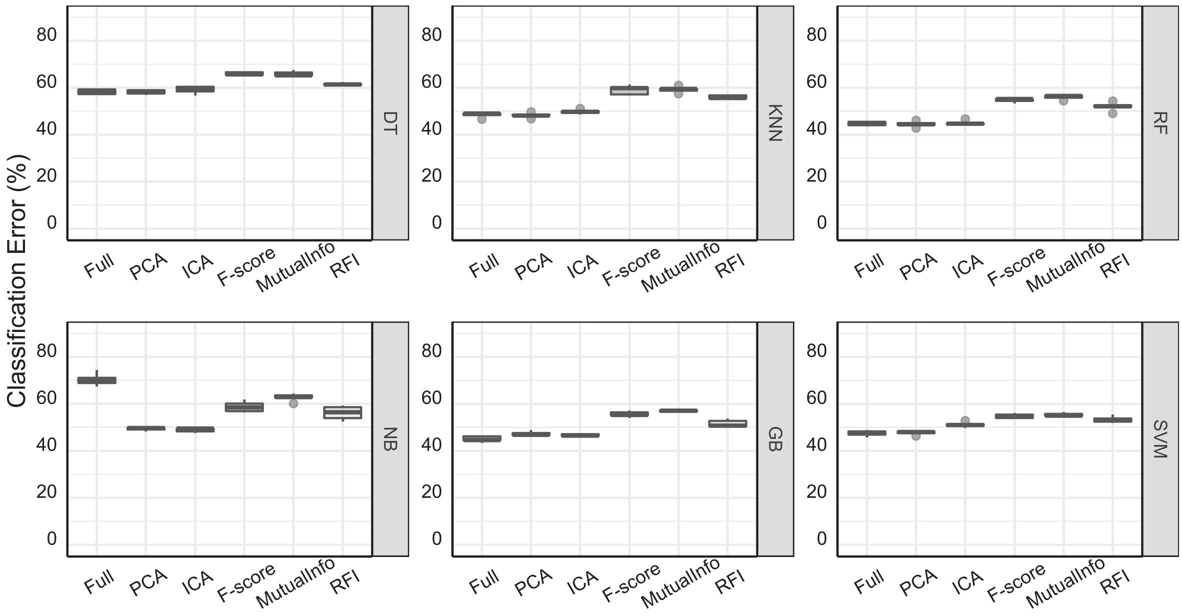
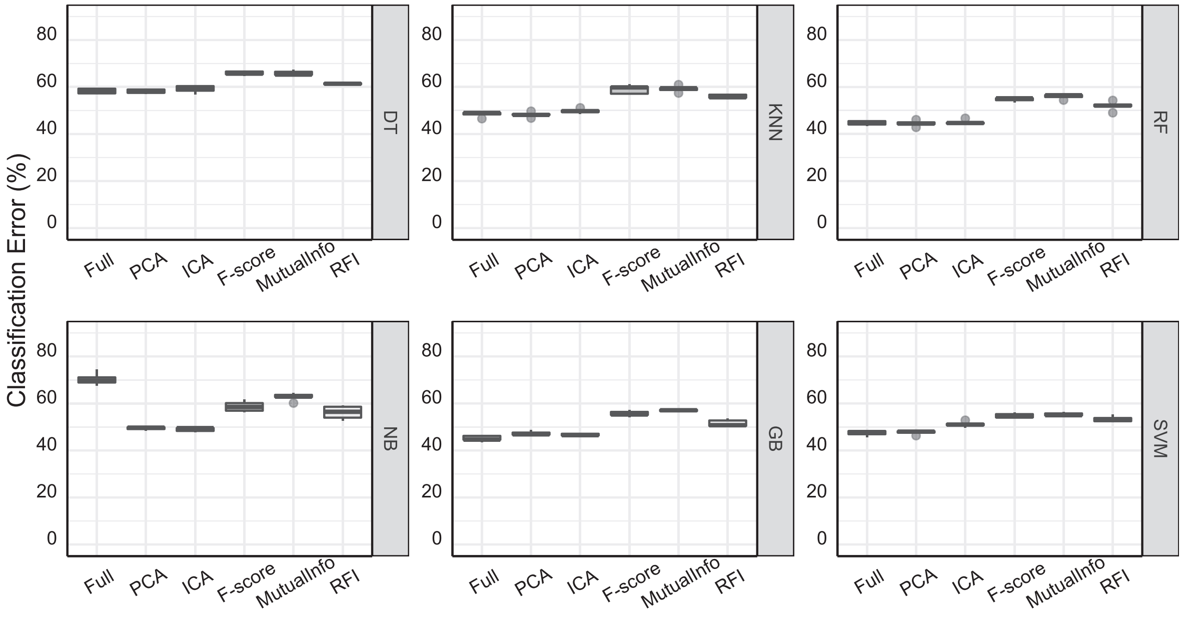


Fig. 10. Percentage of cases where percentage point increase of average CV errors when compared to WSI is statistically larger than values ranging from 0 to 20% for different classifiers.

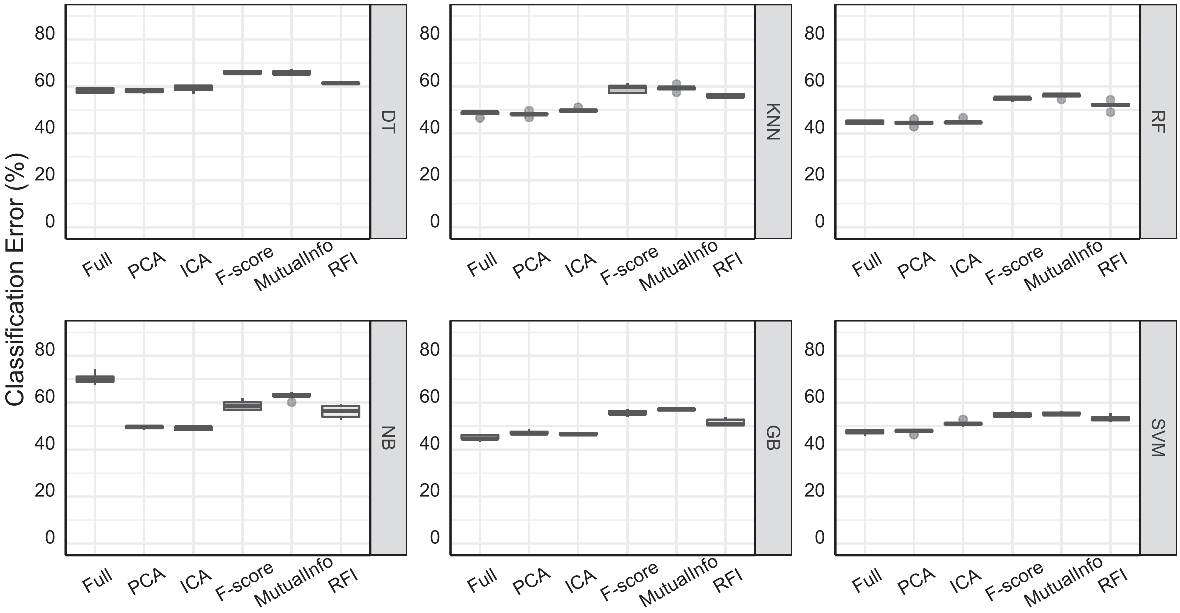


(a) “Airline” dataset

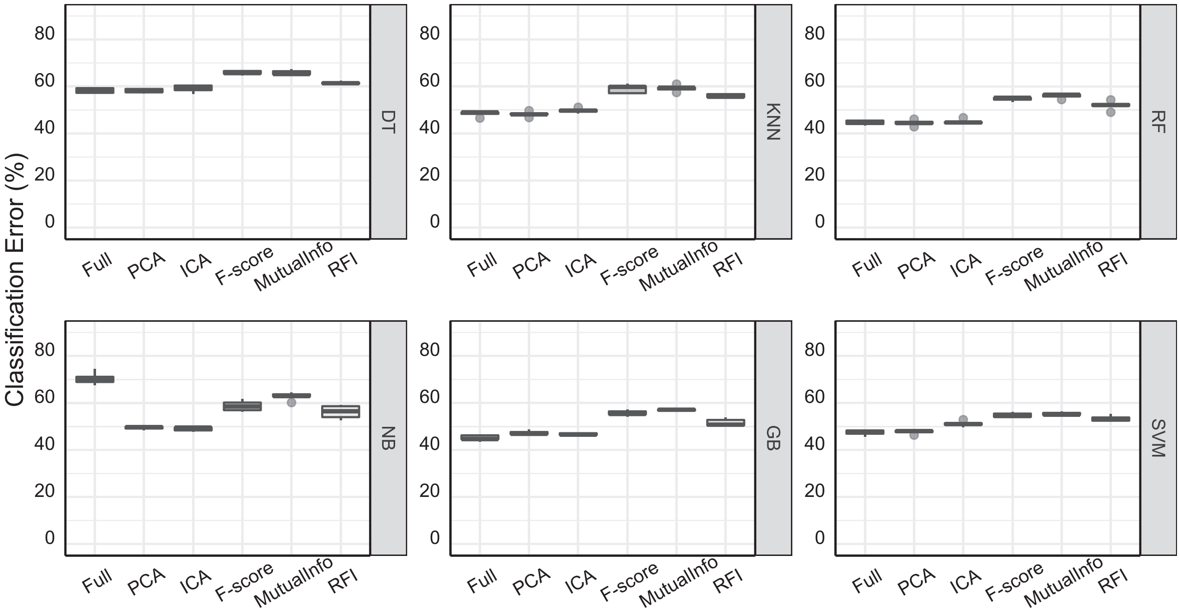


(b) “Connect-4” dataset

Fig. 11. Box plots of mean CV error percentages for different classifiers and feature reduction methods for the case of 2000 randomly selected instances over 25 repetitions.

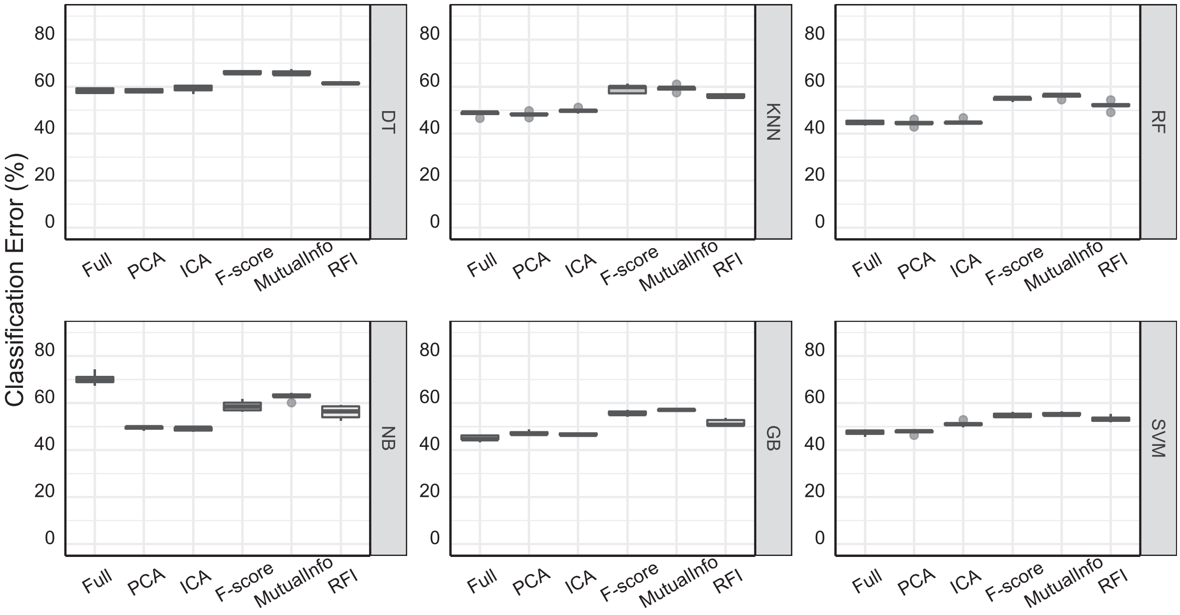


(c) “Covertype” dataset



(d) “Jannis” dataset

Fig. 11. (*continued*).



(e) “MiniBooNE” dataset

Fig. 11. (*continued*).

1000 instances and then performing FR never (statistically) reduces the CV performance by more than 12% and 16% respectively when compared against FR without RIS. However, with 2000 and 1000 random instances, RIS + FR decreases the execution time by a factor of 50 and 65 with mean decrease in CV performance being only 1 and 2 percentage points respectively, again when compared against FR without RIS. On the other hand, selecting 2000 and 1000 random instances and using just 10 features as selected by an FR method result in a dramatic 99.92% and 99.96% reduction in data respectively (across both features and instances) when compared to original dataset sizes with full sets of features and WSI. In addition, there is little variation in performance in the subset of selected instances in our framework and the final performance appears to be robust to the randomness in the selection process. This result is a significant implication in pattern recognition research as it shows that with simple random instance selection followed by FR, the amount of data required for training a classifier can be decreased significantly and therefore classifiers can be trained a lot quicker with minimal impact on classification performance.

At this point, we would like to make it abundantly clear that our results and conclusions are apparently limited to the 20 datasets specifically considered in this study together with the particular parameter values we use for the classifiers and the FR methods that we employ. Nonetheless, we believe that the sheer number of datasets we use and the relatively large size of our datasets (in terms of both number of features and number of instances) together with their diverse application domains are quite encouraging for general applicability of our results across other big data problems.

As future research, one direction would be to investigate the effects of utilizing resampling techniques within the RIS + FR framework. Specifically, under-sampling or over-sampling can be performed prior to random instance selection to mitigate the negative effects of class imbalance issues on prediction performance. Another direction would be to incorporate an optimization algorithm into the instance selection process rather than performing the selection randomly. Since the optimal instance selection problem is inherently noisy and highly nonlinear, black-box stochastic optimization algorithms can perhaps be considered for this task.

One other important direction for future research would be to assess impact of RIS on hyperparameter optimization of a classifier for a given dataset. Hyperparameter optimization is almost always an integral part of predictive modeling as this optimization tends to improve classification performance in general. However, hyperparameter optimization is usually a time-consuming task due to the potentially large number of hyperparameter combinations, which can only be exacerbated by potentially excessive number of instances. Thus, it would be worthwhile to investigate if hyperparameter optimization would return similarly performing models if an instance-reduced dataset is used during the optimization process.

CRediT authorship contribution statement

Milad Malekipirbazari: Data curation, Investigation, Methodology, Formal analysis, Validation, Visualization, Writing - original draft. **Vural Aksakalli:** Conceptualization, Software, Project administration, Resources, Supervision, Writing - review & editing. **Waleed Shafqat:** Data curation, Investigation, Methodology, Validation, Visualization, Writing - original draft. **Andrew Eberhard:** Investigation, Methodology, Project administration, Resources, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 7
Mean CV errors for the largest five datasets with 2000 instances for 25 repetitions of the random selection process. Standard errors are shown with a ± sign.

	Feature Reduction Method	Datasets				
		Airline	Connect-4	Covertype	Jannis	MiniBooNE
DT	F-score	42.89 ± 0.46	41.23 ± 0.54	39.24 ± 0.95	46.83 ± 0.83	16.05 ± 0.31
	MutualInfo	40.75 ± 2.2	35.58 ± 0.7	36.91 ± 1	46.41 ± 1.2	15.3 ± 0.29
	RFI	42.79 ± 0.51	40.96 ± 0.93	38.5 ± 1.02	45.63 ± 0.86	15.62 ± 0.37
	PCA	43.9 ± 0.45	46.2 ± 0.42	39.99 ± 0.57	51.6 ± 0.73	19.36 ± 0.21
	ICA	43.66 ± 0.41	44.32 ± 1.08	40.58 ± 0.61	51.09 ± 0.59	18.57 ± 0.17
	Full	42.44 ± 0.78	38.85 ± 0.76	35.67 ± 0.8	46.76 ± 0.85	14.58 ± 0.5
NB	F-score	39.51 ± 1.16	39.15 ± 1.3	88.02 ± 2.42	44.81 ± 0.8	55.35 ± 7.39
	MutualInfo	50.24 ± 1.54	60.3 ± 10.72	87.58 ± 1.37	44.76 ± 0.62	59.98 ± 8.02
	RFI	38.96 ± 0.54	33.42 ± 0.71	38.68 ± 0.7	43 ± 0.29	60.9 ± 8.2
	PCA	40.73 ± 0.51	34.46 ± 0.34	47.49 ± 0.39	42.65 ± 0.78	25.28 ± 2.27
	ICA	42.28 ± 0.95	34.36 ± 0.49	53.15 ± 0.63	42.21 ± 0.82	40.62 ± 5.64
	Full	52.25 ± 0.42	81.92 ± 2.77	82.4 ± 1.36	51.2 ± 0.63	63.55 ± 5.98
KNN	F-score	40.16 ± 0.38	37.33 ± 0.65	33.24 ± 1.05	43.54 ± 0.5	15.14 ± 0.36
	MutualInfo	43.5 ± 0.86	37.34 ± 0.93	31.68 ± 0.76	42.04 ± 0.46	14.58 ± 0.31
	RFI	40.79 ± 0.3	36.88 ± 1.09	36.88 ± 0.91	41.33 ± 0.9	14.46 ± 0.35
	PCA	41.17 ± 0.5	40.01 ± 0.79	35.62 ± 0.93	46.48 ± 0.69	15.06 ± 0.42
	ICA	40.78 ± 0.51	39.82 ± 1.01	35.93 ± 1.09	48.02 ± 0.79	14.2 ± 0.2
	Full	41.23 ± 0.68	38.64 ± 0.73	33.24 ± 0.83	46.87 ± 1.18	14.97 ± 0.44
GB	F-score	37.7 ± 0.36	30.93 ± 0.85	32.26 ± 0.83	38.41 ± 0.42	11.17 ± 0.4
	RFI	37.73 ± 0.73	30.53 ± 0.74	30.88 ± 0.96	36.36 ± 0.47	11.17 ± 0.15
	PCA	38.73 ± 0.7	35.13 ± 0.47	34.29 ± 0.75	43.1 ± 0.65	13.34 ± 0.23
	ICA	38.32 ± 0.83	34.15 ± 0.76	34.61 ± 0.79	41.91 ± 0.44	12.65 ± 0.3
	MutualInfo	40.01 ± 1.25	31.85 ± 1.06	31.26 ± 0.57	38 ± 0.3	11.29 ± 0.22
	Full	37.72 ± 0.77	27.32 ± 0.64	28.19 ± 0.88	35.86 ± 0.85	8.82 ± 0.3
RF	F-score	41.55 ± 0.59	35.5 ± 0.74	33.81 ± 1.6	38.58 ± 0.71	11.17 ± 0.35
	MutualInfo	42.52 ± 1.71	35.16 ± 0.69	28.85 ± 0.63	37.2 ± 0.48	11.1 ± 0.11
	RFI	38.56 ± 0.74	33.02 ± 0.85	29.3 ± 0.63	36.06 ± 0.85	10.94 ± 0.12
	PCA	39.55 ± 0.68	33.88 ± 0.64	31.75 ± 0.65	42.24 ± 0.91	13.18 ± 0.3
	ICA	38.76 ± 0.54	33.23 ± 0.66	32.39 ± 0.89	41.58 ± 0.54	12.39 ± 0.24
	Full	37.43 ± 0.83	28.6 ± 0.8	26.61 ± 0.57	35.44 ± 0.77	9.39 ± 0.29
SVM	F-score	37.38 ± 0.68	34.27 ± 0.41	32.45 ± 0.78	40.33 ± 0.78	16.52 ± 0.17
	MutualInfo	40.32 ± 1.4	34.22 ± 0.37	33.28 ± 0.92	40.71 ± 0.47	16.56 ± 0.28
	RFI	37.11 ± 0.56	34.15 ± 0.37	35.16 ± 0.66	39.76 ± 0.68	16.62 ± 0.28
	PCA	38.01 ± 0.49	34.18 ± 0.38	40.6 ± 0.63	43.02 ± 0.95	16.13 ± 0.19
	ICA	37.76 ± 0.45	34.19 ± 0.38	43.81 ± 1.13	46.17 ± 0.76	19.18 ± 0.22
	Full	38.76 ± 0.72	34.23 ± 0.35	30.26 ± 0.9	38.22 ± 0.79	16.12 ± 0.18

Acknowledgements

The authors are grateful to the Editor and the two anonymous reviewers for their invaluable comments and suggestions that greatly improved the overall contribution and quality of this manuscript.

References

Aksakalli, V., & Malekipirbazari, M. (2016). Feature selection via binary simultaneous perturbation stochastic approximation. *Pattern Recognition Letters, 75*, 41–47.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., & Herrera, F. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing, 17*.

Arnaiz-González, Á., Díez-Pastor, J.-F., Rodríguez, J. J., & García-Osorio, C. (2016). Instance selection of linear complexity for big data. *Knowledge-Based Systems, 107*, 83–95.

ASU (2015). Arizona State University Feature Selection Repository.

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). Feature selection for high-dimensional data. Springer International Publishing: Imprint: Springer, Cham.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Cover, T. M. (2006). *Elements of information theory*. New Jersey: Wiley-Interscience.

Dosilovic, F., Brcic, M., & Hlupic, N. (2018). Explainable artificial intelligence: A survey. In *Proceedings of 41st international convention on information and communication technology, electronics and microelectronics (MIPRO)*.

Everson, T., Lyons, G., Zhang, H., Soto-Ramirez, N., Lockett, G., Patil, V., Merid, S., Soederhall, C., Melen, E., Holloway, J., Arshad, S., & Karmaus, W. (2015). Dna methylation loci associated with atopy and high serum ige: a genome-wide application of recursive random forest feature selection. *Genome Medicine, 7*(1).

Fernandes, K., Vinagre, P., & Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. In *Portuguese conference on artificial intelligence* (pp. 535–546). Springer.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*(2), 179–188.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), 367–378.

Guyon, I., Gunn, S., Ben-Hur, A., & Dror, G. (2005). Result analysis of the nips 2003 feature selection challenge. *Advances in Neural Information Processing Systems*, 545–552.

Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO), IEEE, pp. 1200–1205.

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. Cambridge, Massachusetts: The MIT Press.

Lichman, M. (2013). UCI machine learning repository.

Liu, H., & Motoda, H. (2002). On issues of instance selection. *Data Mining and Knowledge Discovery, 6*(2), 115.

Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems, 62*, 22–31.

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Kittler, J. (2010). A review of instance selection methods. *Artificial Intelligence Review, 34*(2), 133–143.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Pierce, K. M., Hope, J. L., Johnson, K. J., Wright, B. W., & Synovec, R. E. (2005). Classification of gasoline data obtained by gas chromatography using a piecewise alignment algorithm combined with feature selection and principal component analysis. *Journal of Chromatography A, 1096*(1), 101–110.

Polat, K., & Günes, S. (2009). A new feature selection method on classification of medical datasets: Kernel f-score feature selection. *Expert Systems With Applications, 36*(7), 10367–10373.

Rong, M., Gong, D., & Gao, X. (2019). Feature selection and its use in big data: Challenges, methods, and trends. *IEEE Access, 7*, 19709–19725.

Senawi, A., Wei, H., & Billings, S. (2017). A new maximum relevance-minimum multicollinearity (mrmmc) method for feature selection and ranking. *Pattern Recognition, 67*, 47–61.

Sikonia, M., & Kononenko, I. (2003). Theoretical and empirical analysis of relief and relieff. *Machine Learning, 53*(23–69).

Song, Y., Liang, J., Lu, J., & Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing, 251*, 26–34.

Stone, J. V. (2004). *Independent component analysis: a tutorial introduction*. MIT press.

Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). Openml: Networked science in machine learning. *SIGKDD Explorations, 15*(2), 49–60.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1* (6), 80–83.

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning, 38*(3), 257–286.

Wong, T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition, 48*(9), 2839–2846.

Yeh, I.-C., & Lien, C.-H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications, 36*(2), 2473–2480.

Zheng, Z., Chenmao, X., & Jia, J. (2010). Iso-container projection for feature extraction. In *Proceedings of IEEE international symposium on intelligent signal processing and communication systems*.