basic.png

Figure 1: Using imputation for classification with incomplete data.

Improving Classification with Incomplete Data Using Feature Selection and Clustering

Cao Truong Tran[a,b,*], Mengjie Zhang[a], Peter Andreae[a], Bing Xue[a], Lam Thu Bui[b]

[a] *School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand*
[b] *Le Qui Don Technical University, 236 Hoang Quoc Viet St, Hanoi, Vietnam.*

Abstract

Keywords: missing data, classification, genetic programming

1. Introduction

2. Related Work

2.1. Imputation

2.1.1. kNN-based Imputation

2.1.2. Multiple Imputation by Chained Equations

2.2. Imputation for Classification with Incomplete Data

Figure. 1 shows main steps of using imputation for classification with incomplete data.

2.3. Clustering

2.4. Feature selection

2.5. Differential Evolution

2.6. DE for feature selection

3. The Proposed Method

Three combinations of imputation, feature selection and clustering are proposed to improve classification with incomplete data. The first combination is between imputation and feature selection. The second combination is between imputation and clustering. The third combination is between imputation, feature selection and clustering. Each of the three combinations includes a training process and an application process. The training process uses a training data to build a classifier which is used to classify a new instance in the application process.

*Corresponding author. Tel: +64221587242
Email address: cao.truong.tran@ecs.vuw.ac.nz (Cao Truong Tran)

Cl.png

Figure 2: Combining imputation and clustering for classification with incomplete data.

3.1. The combination of Imputation and Clustering

The key idea of the combination of imputation and clustering for classification with incomplete data is to use clustering to reduce the number of instances in the training imputed data. After that, in the application process, incomplete instances are estimated missing values based on a smaller training data. Consequently, the computation time to estimate missing values in the application process could be reduced.

Figure. 2 shows main steps of the combination of imputation and clustering for classification with incomplete data. In the training process, an imputation method is used to estimate missing values in the incomplete training data. Subsequently, on the one hand, the imputed training data is used by an classification algorithm to build a classifier. One the other hand, the training imputed data is put into a clustering algorithm to construct a training clustered data which will be used to estimate missing values in the application process. In the application process, when a new instance need to be classified, if it is complete, it will be directly classified by the classifier. Otherwise, it will be combined with the clustered training data, and then put into an the imputation method to estimate missing values. Afterwards, the imputed instance is classified by the classifier.

3.2. The combination of Imputation and Feature Selection

The key idea of the combination of imputation and feature selection for classification with incomplete data is to use feature selection to remove redundant

Fs.png

Figure 3: Combining imputation and feature selection for classification with incomplete data.

features in the training imputed data. As a result, feature selection can improve the quality of training data which helps to construct better classifiers. Moreover, by removing redundant features, feature selection can not only generate smaller training data but also reduce the number of incomplete instances in the testing data. Consequently, the computation time for estimating missing values in the testing process could be reduced.

Figure. 3 shows main steps of the combination of imputation and feature selection for classification with incomplete data. In the training process, first an imputation method is used to estimate missing values in the training incomplete data. After that, a feature selection method is used to remove redundant features in the training imputed data. The training selected data is then put into a classification algorithm to build a classifier. In the application process, when a new instance need to be classified, first redundant values in the instance are removed by only keeping values in selected features. Thereafter, if the selected instance is complete, it is directly classified by the classifier. Otherwise, missing values in the instance are estimated by using the imputation method and the training selected data. Subsequently, the imputed instance is classified by the classifier.

3.3. The combination of Imputation, Feature Selection and Clustering

The key idea of the combination of imputation, feature selection and clustering for classification with incomplete data is that using both feature selection

FsCl.png

Figure 4: Combining imputation, feature selection and clustering for classification with incomplete data.

and clustering not only can remove redundant features, but also can reduce the number of instances in the imputed training data. As a result, by removing redundant features, feature selection can improve classification accuracy, and reduce the computation time to estimate missing values in the application process, simultaneously. In addition, by reducing the number of instances in the imputed data, clustering can further reduce the computation time to estimate missing values in the application process.

Figure. 4 shows main steps of the combination of imputation, feature selection and clustering for classification with incomplete data. In the training process, firstly, the training incomplete data is put into an imputation method to estimate missing values. After that, the training imputed data is put into a feature selection method to remove redundant features. Following that, one the one hand, the training selected data is used by a classification to build a classifier. On the other hand, the training selected data is used by a clustering method to generate a smaller training data which is then used to estimate missing values in the application process. In the application process, when a new instance need to be classified, firstly, redundant values of the instance are eliminated by only keeping values in the selected features. Subsequently, if the selected instance is complete, it will be directly classified by the classifier. Otherwise, it is combined with the clustered data, and then is used by the imputation method to estimate missing values. Finally, the imputed instance is classified by the classifier.

4. Experiment Design

This section presents the comparison method, datasets used in experiments and parameter settings.

4.1. The Comparison Method

Experiments were designed to evaluate the effectiveness of the proposed methods. To evaluate the combination of imputation and clustering shown in Fig.2, it is compared with the method only using imputation shown in Fig.1. To evaluate the combination of imputation and clustering shown in Fig.3, it is compared with the method only using imputation shown in Fig.1. To evaluate the combination of imputation, feature selection and clustering as shown in Fig.4, it is compared with the method only using imputation shown in Fig.1, the combination of imputation and clustering shown in Fig.2, and the combination of imputation and feature selection shown in Fig.3.

4.2. Datasets

The proposed methods are tested in ten incomplete datasets. The datasets are chosen from the the UCI Machine Learning Repository [3]. The Table 1 shows the main characteristics of the chosen datasets including the number of instances, the number of features (R/I/N: Real/Integer/Nominal), the number of classes, and the percentage of incomplete instances.

The datasets are carefully chosen to include a different collection of problem domains. The datasets have varying percentages of incomplete instances (incomplete instances range between 5% and 100% of total instances). The datasets also range from large number of instances (Mar has 8993 instances) to small number of instances (Hep only has 155 instances). The datasets also range between high and low dimensionality (Arr has 279 features while Mam only has 4 features). The datasets also have varying types of features including real, integer and nominal. It is to be hoped that the datasets can reflect incomplete problems of varying difficulty, size, dimensionality and feature types.

Table 1: Datasets used in the experiments

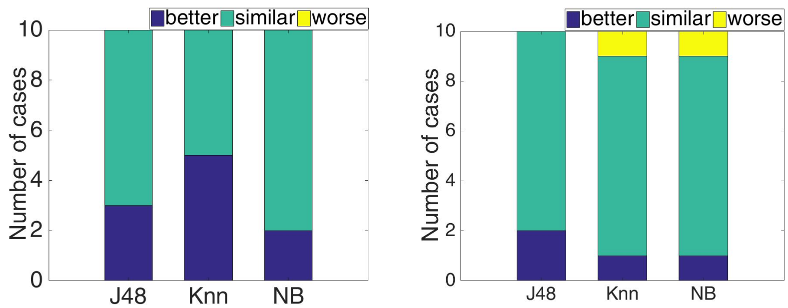
Name	**#Inst**	**#Features (R/I/N)**	**#Classes**	**Incomplete inst(%)**	**Abbrev**
Arrhythmia	452	279(206/0/73)	16	85.11	Arr
Automobile	205	25(15/0/10)	6	26.83	Aut
Credit Approval	690	15(3/3/9)	2	5.36	Cre
Heart Disease	303	13(0/0/13)	5	100	Hea
Hepatitis	155	19(2/17/0)	2	48.39	Hep
Horse-colic	368	23(7/1/15)	2	98.1	Hor
Housevotes	435	16(0/0/16)	2	46.67	Hou
Mammographic	961	5(0/5/0)	2	13.63	Mam
Marketing	8993	13(0/13/0)	9	23.54	Mar
Ozone	2536	73(73/0/0)	2	27.12	Ozo

None of the datasets is divided into a training set and a test set. Moreover, the number of instances in some datasets are relatively small. Therefore, ten-fold cross-validation method is used to divide the datasets into training and test datasets. Furthermore, ten-fold cross-validation process is stochastic, so it should be performed multiple times. In the experiments, for each dataset , ten-fold cross-validation is performed 30 times. As a result, 300 pairs of training and test datasets are generated from one dataset.

4.3. Parameter Settings

4.3.1. Imputation Methods

The experiments use two imputation methods: Knn-based imputation and MICE imputation. These imputation methods are selected to represent two categories of imputation methods: single imputation and multiple imputation, respectively. Knn-based imputation with K=1 is used since it is simple, fast and non-parametric. Multivariate imputation by chained equations in R [22] is used for MICE's implementation. In MICE, random forest [46] is used as a regression method to estimate missing values. Each incomplete feature is



(a) Using Knn-based imputation

(b) Using Mice imputation

Figure 5: Comparison between the combination of imputation with clustering and only using imputation.

4.3.2. Clustering

A k-means^{++} clustering algorithm [?] is used to cluster data. WEKA [47] is used to implement the clustering algorithm. The number of K in the clustering algorithm is set as a square root of the number of instances.

4.3.3. DE for Feature Selection

The parameters of the DE based algorithms are set as follows. The population size is 30 and the maximum number of generations is 50. The mutation factor is set as 1. The crossover rate is set as 0.25. The threshold θ is set as 0.6.

4.3.4. Classification algorithms

The experiments use three classification algorithms: C4.5, kNN and Naive-Bayes. These classification algorithms are selected to represent three categories of classifiers: rule-based learning, lazy learning and approximate models, respectively. WEKA [47] is used to implement the classification algorithms.

5. Results and Analysis

5.1. Classification Accuracy

5.1.1. Imputation Combined Clustering

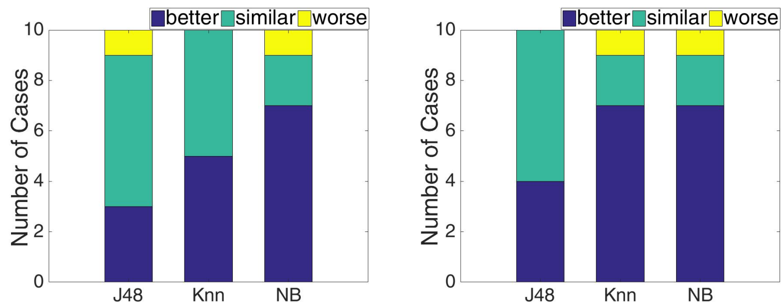
5.1.2. Imputation Combined Feature Selection

Table 2: Using Knn-based imputation

Dataset	**Classifier**	**Knn**	**KnnCl**	**T**	**KnnFs**	**T**	**KnnFsCl**	**T**
Arr	J48	64.59±3.13	65.38±1.54	=	66.07±1.61	=	**66.16±1.65**	=
	Knn	58.19±0.83	58.21±0.82	+	60.41±1.06	+	**60.47±1.14**	+
	NB	61.32±1.42	61.26±1.47	=	**64.19±1.85**	+	**64.19±1.77**	+
Aut	J48	66.84±3.92	**66.91±3.75**	+	64.63±3.91	-	64.74±4.33	-
	Knn	53.12±3.39	53.07±3.45	+	**54.91±3.01**	=	54.66±3.01	=
	NB	53.65±3.37	53.58±3.21	=	**60.05±4.53**	+	59.83±4.46	+
Cre	J48	85.09±0.63	**85.20±0.60**	+	84.84±0.74	=	84.82±0.73	=
	Knn	85.90±0.58	**86.07±0.53**	+	85.58±0.71	=	85.57±0.71	=
	NB	77.36±0.43	77.35±0.42	=	**87.04±0.47**	+	87.03±0.48	+
Hea	J48	78.67±1.50	78.75±1.46	=	79.85±1.33	+	**79.89±1.27**	+
	Knn	79.60±0.95	**80.03±1.15**	+	78.76±2.06	=	79.04±2.19	=
	NB	**82.42±0.75**	82.31±0.59	=	79.50±1.51	-	79.67±1.33	-
Hep	J48	78.48±2.08	78.78±2.31	=	79.04±2.28	=	**79.47±2.18**	+
	Knn	81.51±1.08	81.72±1.21	=	82.13±2.40	=	**82.50±2.34**	=
	NB	84.02±0.88	**84.59±0.84**	+	80.89±1.63	-	81.33±1.65	-
Hor	J48	83.74±0.86	83.67±0.96	=	83.81±0.98	=	**83.98±0.93**	=
	Knn	78.82±1.33	78.67±1.17	=	83.08±0.96	+	**83.23±1.20**	+
	NB	75.96±0.65	75.86±0.88	=	82.37±1.04	+	**82.48±1.03**	+
Hou	J48	96.24±0.62	96.29±0.58	=	96.26±0.66	=	**96.33±0.61**	=
	Knn	93.68±0.39	93.69±0.49	=	**94.59±0.70**	+	94.47±0.73	+
	NB	90.20±0.24	90.14±0.31	=	95.10±0.60	+	**95.14±0.63**	+
Mam	J48	81.99±0.58	81.91±0.63	=	**82.18±0.61**	=	82.12±0.59	=
	Knn	78.66±0.59	78.62±0.64	=	**82.74±0.63**	+	82.56±0.69	+
	NB	80.63±0.51	80.69±0.40	=	80.72±0.55	=	**80.82±0.53**	=
Mar	J48	29.90±0.41	29.89±0.48	=	**32.60±0.43**	+	32.56±0.40	+
	Knn	28.24±0.36	28.25±0.37	=	**32.10±0.52**	+	**32.10±0.51**	+
	NB	30.53±0.32	30.54±0.32	=	**32.21±0.30**	+	32.17±0.31	+
Ozo	J48	95.74±0.79	95.93±0.37	+	96.39±0.77	+	**96.54±0.35**	+
	Knn	96.69±0.27	**96.79±0.15**	+	96.63±0.29	=	96.74±0.16	=
	NB	70.89±1.42	73.06±1.78	+	97.01±0.13	+	**97.03±0.10**	+

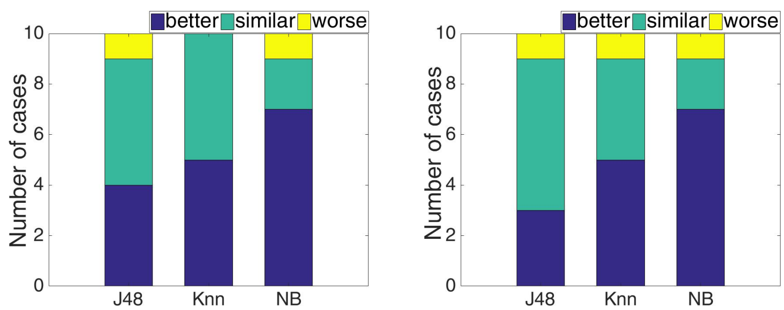
Table 3: Using Mice imputation

Dataset	Classifier	Mice	MiceCl	T	MiceFs	T	MiceFsCl	T
Arr	J48	65.44±1.66	65.44±1.66	=	**65.77±2.16**	=	**65.77±2.16**	=
	Knn	59.04±0.85	59.07±0.80	=	**61.08±1.23**	+	**61.08±1.24**	+
	NB	62.13±1.24	62.12±1.21	=	**64.66±1.76**	+	64.59±1.78	+
Aut	J48	**67.89±4.24**	67.62±4.41	=	66.04±4.50	=	65.64±4.47	-
	Knn	57.51±2.34	57.51±2.35	=	**59.14±3.52**	+	58.97±3.61	=
	NB	56.07±3.59	55.68±3.66	=	**60.04±2.73**	+	59.80±2.70	+
Cre	J48	85.30±0.58	**85.32±0.61**	+	85.15±0.66	=	85.15±0.65	=
	Knn	85.96±0.56	**86.04±0.55**	=	85.05±0.76	-	85.04±0.80	-
	NB	77.26±0.44	77.25±0.47	=	**86.87±0.49**	+	86.86±0.50	+
Hea	J48	78.36±1.29	78.99±1.22	+	79.39±1.14	+	**79.51±1.24**	+
	Knn	**81.64±1.07**	81.16±1.70	=	78.04±1.38	-	77.60±1.62	-
	NB	**82.79±0.43**	82.78±0.42	=	81.44±0.80	-	81.57±0.70	-
Hep	J48	80.01±2.25	79.73±2.05	=	**81.68±2.11**	+	80.70±2.56	=
	Knn	82.27±1.27	82.25±1.35	=	**83.62±2.26**	+	83.34±2.26	=
	NB	84.27±0.82	84.20±0.85	=	82.97±1.84	-	82.73±1.66	-
Hor	J48	84.26±0.78	**84.55±0.41**	=	84.02±0.97	=	83.99±0.97	=
	Knn	78.95±1.17	78.31±1.27	-	83.58±1.12	+	**83.68±1.24**	+
	NB	77.51±0.75	76.39±0.82	-	**81.35±1.29**	+	80.97±1.55	+
Hou	J48	96.15±0.54	96.15±0.60	=	**96.22±0.57**	=	**96.22±0.56**	=
	Knn	93.84±0.31	93.85±0.36	+	**94.51±0.54**	+	94.49±0.54	+
	NB	91.11±0.20	91.11±0.23	=	**95.72±0.43**	+	**95.72±0.43**	+
Mam	J48	82.24±0.65	82.15±0.57	=	**82.86±0.50**	+	82.80±0.55	+
	Knn	78.52±0.67	78.55±0.62	=	**83.03±0.46**	+	82.93±0.44	+
	NB	**80.73±0.37**	80.64±0.42	=	80.47±0.76	=	80.33±0.80	-
Mar	J48	30.01±0.45	29.98±0.44	=	**32.60±0.45**	+	32.59±0.42	+
	Knn	28.20±0.30	28.23±0.32	=	**31.95±0.33**	+	**31.95±0.34**	+
	NB	30.56±0.31	30.58±0.30	=	32.42±0.29	+	**32.44±0.30**	+
Ozo	J48	95.89±0.41	95.88±0.42	=	**96.12±0.42**	=	**96.12±0.42**	=
	Knn	96.77±0.16	**96.79±0.17**	=	**96.79±0.12**	=	96.78±0.13	=
	NB	71.46±0.44	72.27±0.50	+	**96.18±1.16**	+	96.16±1.18	+



(a) Using Knn-based imputation (b) Using Mice imputation

Figure 6: Comparison between the combination of imputation with clustering and only using imputation.



(a) Using Knn-based imputation (b) Using Mice imputation

Figure 7: Comparison between the combination of imputation with clustering and only using imputation.

5.1.3. Imputation Combined Feature Selection and Clustering

5.2. Computation Time

5.2.1. Knn-based Imputation

5.2.2. Mice Imputation

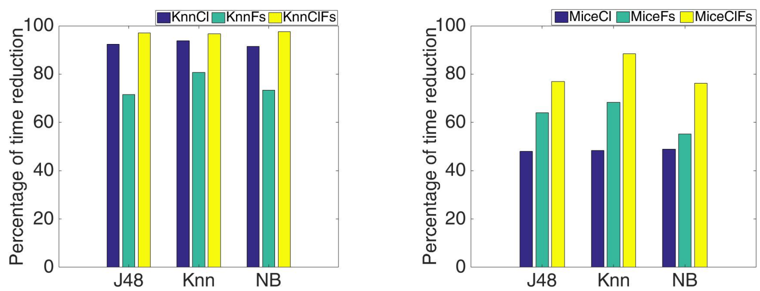
6. Conclusion

References

[1] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, John Wiley & Sons, 2012.

Table 4: Computation time

MiceFs	MiceFsCl	KnnFsCl	KnnFs	Mice	MiceCl	KnnCl	Knn
3.13	3.71	3.76	4.01	4.60	4.76	5.63	6.36



(a) Using Knn-based imputation (b) Using Mice imputation

Figure 8: Comparison between the combination of imputation with clustering and only using imputation.

[2] P. J. García-Laencina, J.-L. Sancho-Gómez, A. R. Figueiras-Vidal, Pattern classification with missing data: a review, Neural Computing and Applications 19 (2010) 263–282.

[3] M. Lichman, UCI machine learning repository (2013).
URL `http://archive.ics.uci.edu/ml`

[4] R. J. Little, D. B. Rubin, Statistical analysis with missing data, John Wiley & Sons, 2014.

[5] A. Farhangfar, L. A. Kurgan, W. Pedrycz, A novel framework for imputation of missing values in databases, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 37 (2007) 692–709.

[6] E.-L. Silva-Ramírez, R. Pino-Mejías, M. López-Coello, Single imputation with multilayer perceptron and multiple imputation combining multilayer perceptron and k-nearest neighbours for monotone patterns, Applied Soft Computing 29 (2015) 65–74.

Table 5: Computation time

Data	Class	Knn-based imputation				Mice imputation			
		Knn	KnnCl	KnnFs	KnnFsCl	Mice	MiceCl	MiceFs	MiceFsCl
Arr	J48	1.7×10^{2}	1.3×10^{1}	4.1×10^{1}	2.9×10^{0}	4.9×10^{7}	4.3×10^{7}	3.7×10^{5}	3.3×10^{5}
	Knn	1.5×10^{2}	1.1×10^{1}	2.9×10^{1}	2.1×10^{0}	4.9×10^{7}	4.2×10^{7}	4.4×10^{5}	4.1×10^{5}
	NB	1.6×10^{2}	1.2×10^{1}	3.3×10^{1}	2.5×10^{0}	4.8×10^{7}	4.7×10^{7}	3.5×10^{5}	3.2×10^{5}
Aut	J48	9.3×10^{-1}	1.6×10^{-1}	4.3×10^{-1}	1.0×10^{-1}	1.3×10^{5}	1.0×10^{5}	7.3×10^{4}	6.4×10^{4}
	Knn	8.5×10^{-1}	1.8×10^{-1}	3.6×10^{-1}	1.5×10^{-1}	6.8×10^{4}	5.0×10^{4}	4.0×10^{4}	3.1×10^{4}
	NB	3.3×10^{-1}	6.7×10^{-2}	1.3×10^{-1}	1.6×10^{-2}	7.1×10^{4}	4.9×10^{4}	3.7×10^{4}	3.1×10^{4}
Cre	J48	7.5×10^{-1}	4.0×10^{-2}	2.3×10^{-1}	1.6×10^{-2}	2.7×10^{4}	1.1×10^{4}	1.2×10^{4}	2.8×10^{3}
	Knn	6.4×10^{-1}	2.3×10^{-2}	1.5×10^{-1}	6.6×10^{-3}	1.2×10^{4}	5.1×10^{3}	5.7×10^{3}	1.3×10^{2}
	NB	5.6×10^{-1}	3.7×10^{-2}	2.4×10^{-1}	1.6×10^{-2}	1.3×10^{4}	5.1×10^{3}	2.3×10^{3}	1.2×10^{3}
Hea	J48	1.5×10^{0}	1.3×10^{-1}	7.8×10^{-1}	5.0×10^{-2}	1.4×10^{5}	8.5×10^{4}	9.2×10^{4}	6.6×10^{4}
	Knn	1.5×10^{0}	1.1×10^{-1}	8.6×10^{-1}	8.6×10^{-2}	7.2×10^{4}	4.5×10^{4}	3.6×10^{4}	1.3×10^{2}
	NB	1.6×10^{0}	1.3×10^{-1}	9.4×10^{-1}	9.6×10^{-2}	7.5×10^{4}	4.7×10^{4}	5.7×10^{4}	4.0×10^{4}
Hep	J48	3.2×10^{-1}	5.0×10^{-2}	7.0×10^{-2}	1.3×10^{-2}	3.5×10^{4}	2.5×10^{4}	2.2×10^{4}	1.8×10^{4}
	Knn	3.3×10^{-1}	2.7×10^{-2}	9.0×10^{-2}	2.3×10^{-2}	1.8×10^{4}	1.2×10^{4}	2.7×10^{3}	2.2×10^{3}
	NB	2.8×10^{-1}	4.3×10^{-2}	1.1×10^{-1}	1.3×10^{-2}	1.8×10^{4}	1.4×10^{4}	1.3×10^{4}	9.4×10^{3}
Hor	J48	4.6×10^{0}	3.1×10^{-1}	1.5×10^{0}	1.3×10^{-1}	2.5×10^{5}	1.5×10^{5}	1.1×10^{5}	6.5×10^{4}
	Knn	4.9×10^{0}	4.1×10^{-1}	1.3×10^{0}	1.2×10^{-1}	1.3×10^{5}	7.6×10^{4}	1.3×10^{4}	9.2×10^{3}
	NB	4.4×10^{0}	3.7×10^{-1}	8.9×10^{-1}	7.6×10^{-2}	1.4×10^{5}	7.6×10^{4}	5.7×10^{4}	3.6×10^{4}
Hou	J48	2.3×10^{0}	1.8×10^{-1}	7.1×10^{-1}	4.6×10^{-2}	3.7×10^{5}	2.2×10^{5}	4.1×10^{4}	2.5×10^{4}
	Knn	2.5×10^{0}	1.6×10^{-1}	3.0×10^{-1}	2.3×10^{-2}	1.7×10^{5}	1.0×10^{5}	1.9×10^{3}	1.3×10^{3}
	NB	2.4×10^{0}	1.9×10^{-1}	2.5×10^{-1}	2.3×10^{-2}	1.7×10^{5}	7.4×10^{4}	9.9×10^{4}	6.5×10^{3}
Mam	J48	2.2×10^{0}	6.0×10^{-2}	6.8×10^{-1}	3.0×10^{-2}	1.1×10^{5}	4.3×10^{4}	9.1×10^{4}	3.8×10^{4}
	Knn	1.2×10^{0}	2.6×10^{-2}	2.7×10^{-1}	1.6×10^{-2}	5.1×10^{4}	2.2×10^{4}	4.9×10^{4}	2.1×10^{4}
	NB	1.1×10^{0}	6.3×10^{-2}	1.3×10^{-1}	6.6×10^{-3}	4.8×10^{4}	1.9×10^{4}	4.3×10^{4}	1.9×10^{4}
Mar	J48	8.5×10^{2}	1.3×10^{1}	1.1×10^{2}	1.8×10^{0}	5.9×10^{7}	3.6×10^{6}	6.7×10^{6}	9.3×10^{5}
	Knn	9.3×10^{2}	1.4×10^{1}	9.2×10^{1}	1.5×10^{0}	3.1×10^{7}	1.9×10^{6}	3.1×10^{6}	3.8×10^{5}
	NB	8.5×10^{2}	1.3×10^{1}	1.4×10^{2}	2.6×10^{0}	3.1×10^{7}	1.8×10^{6}	3.1×10^{6}	3.7×10^{5}
Ozo	J48	3.9×10^{2}	1.2×10^{1}	1.2×10^{1}	3.2×10^{0}	3.4×10^{7}	5.8×10^{6}	7.2×10^{6}	1.3×10^{6}
	Knn	3.7×10^{2}	1.2×10^{1}	1.5×10^{1}	4.5×10^{0}	1.8×10^{7}	3.3×10^{6}	4.8×10^{6}	8.9×10^{5}
	NB	3.6×10^{2}	1.1×10^{1}	2.4×10^{1}	7.1×10^{-1}	1.8×10^{7}	3.7×10^{6}	5.4×10^{6}	1.4×10^{6}

[7] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognition 41 (2008) 3692–3705.

[8] F. Lobato, C. Sales, I. Araujo, V. Tadaiesky, L. Dias, L. Ramos, A. Santana, Multi-objective genetic algorithm for missing data imputation, Pattern Recognition Letters 68 (2015) 126–131.

[9] I. R. White, P. Royston, A. M. Wood, Multiple imputation using chained equations: issues and guidance for practice, Statistics in medicine 30 (2011) 377–399.

[10] C. T. Tran, M. Zhang, P. Andreae, A genetic programming-based imputation method for classification with missing data, in: European Conference on Genetic Programming, 2016, pp. 149–163.

[11] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, A. Bouras, A survey of clustering algorithms for big data: Taxonomy and empirical analysis, IEEE transactions on emerging topics in computing 2 (2014) 267–279.

[12] A. Jose-Garcia, W. Gomez-Flores, Automatic clustering using nature-inspired metaheuristics: A survey, Applied Soft Computing 41 (2016) 192–213.

[13] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Computers & Electrical Engineering 40 (2014) 16–28.

[14] B. Xue, M. Zhang, W. N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation 20 (2016) 606–626.

[15] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization 11 (1997) 341–359.

[16] R. N. Khushaba, A. Al-Ani, A. Al-Jumaily, Feature subset selection using differential evolution and a statistical repair mechanism, Expert Systems with Applications 38 (2011) 11515–11526.

[17] A. Al-Ani, A. Alsukker, R. N. Khushaba, Feature subset selection using differential evolution and a wheel based search strategy, Swarm and Evolutionary Computation 9 (2013) 15–26.

[18] A. Ghosh, A. Datta, S. Ghosh, Self-adaptive differential evolution for feature selection in hyperspectral image data, Applied Soft Computing 13 (2013) 1969–1977.

[19] B. Xue, W. Fu, M. Zhang, Multi-objective feature selection in classification: A differential evolution approach., in: SEAL, 2014, pp. 516–528.

[20] G. E. Batista, M. C. Monard, et al., A study of k-nearest neighbour as an imputation method., HIS 87 (2002) 251–260.

[21] E. Acuna, C. Rodriguez, The treatment of missing values and its effect on classifier accuracy, Classification, clustering, and data mining applications (2004) 639–647.

[22] S. Buuren, K. Groothuis-Oudshoorn, mice: Multivariate imputation by chained equations in R, Journal of statistical software 45.

[23] P. Royston, I. R. White, et al., Multiple imputation by chained equations (mice): implementation in stata, Journal of Statistical Software 45 (2011) 1–20.

[24] J. Luengo, S. García, F. Herrera, On the choice of the best imputation methods for missing values considering three groups of classification methods, Knowledge and information systems 32 (2012) 77–108.

[25] G. E. Batista, M. C. Monard, An analysis of four missing data treatment methods for supervised learning, Applied artificial intelligence 17 (2003) 519–533.

[26] Y. Liu, S. D. Brown, Comparison of five iterative imputation methods for multivariate classification, Chemometrics and Intelligent Laboratory Systems 120 (2013) 106–115.

[27] C. T. Tran, M. Zhang, P. Andreae, B. Xue, L. T. Bui, Multiple imputation and ensemble learning for classification with incomplete data, in: Intelligent and Evolutionary Systems: The 20th Asia Pacific Symposium, IES 2016, Canberra, Australia, November 2016, Proceedings, 2017, pp. 401–415.

[28] B. Xue, M. Zhang, Evolutionary feature manipulation in data mining/big data, ACM SIGEVOlution 10 (2017) 4–11.

[29] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, A. Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, IEEE transactions on pattern analysis and machine intelligence 24 (2002) 881–892.

[30] D. Li, J. Deogun, W. Spaulding, B. Shuart, Towards missing data imputation: a study of fuzzy k-means clustering method, in: Rough sets and current trends in computing, Vol. 3066, 2004, pp. 573–579.

[31] C. Zhang, Y. Qin, X. Zhu, J. Zhang, S. Zhang, Clustering-based missing value imputation for data preprocessing, in: Industrial Informatics, 2006 IEEE International Conference on, 2006, pp. 1081–1086.

[32] B. M. Patil, R. C. Joshi, D. Toshniwal, Missing value imputation based on k-mean clustering with weighted distance, in: International Conference on Contemporary Computing, 2010, pp. 600–609.

[33] S. Gajawada, D. Toshniwal, Missing value imputation method based on clustering and nearest neighbours, International Journal of Future Computer and Communication 1 (2012) 206.

[34] J. Tian, B. Yu, D. Yu, S. Ma, Clustering-based multiple imputation via gray relational analysis for missing data and its application to aerospace field, The Scientific World Journal 2013.

[35] Y. UshaRani, P. Sammulal, A novel approach for imputation of missing attribute values for efficient mining of medical datasets-class based cluster approach, arXiv preprint arXiv:1605.01010.

[36] C.-F. Tsai, F.-Y. Chang, Combining instance selection for better missing value imputation, Journal of Systems and Software 122 (2016) 63 – 71.

[37] P. Meesad, K. Hengpraprohm, Combination of knn-based feature selection and knn-based missing-value imputation of microarray data, in: Innovative Computing Information and Control, 2008. ICICIC'08. 3rd International Conference on, 2008, pp. 341–341.

[38] A. Aussem, S. R. de Morais, A conservative feature subset selection algorithm with missing data, Neurocomputing 73 (2010) 585–590.

[39] Q. Lou, Z. Obradovic, Margin-based feature selection in incomplete data., in: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012, pp. 1040–1046.

[40] G. Doquire, M. Verleysen, Feature selection with missing data using mutual information estimators, Neurocomputing 90 (2012) 3–11.

[41] W. Qian, W. Shu, Mutual information criterion for feature selection from incomplete data, Neurocomputing 168 (2015) 210–220.

[42] Q. Long, B. A. Johnson, Variable selection in the presence of missing data: resampling and imputation, Biostatistics 16 (2015) 596–610.

[43] C. T. Tran, M. Zhang, P. Andreae, B. Xue, Improving performance for classification with incomplete data using wrapper-based feature selection, Evolutionary Intelligence 9 (2016) 81–94.

[44] C. T. Tran, M. Zhang, P. Andreae, B. Xue, Bagging and feature selection for classification with incomplete data, in: European Conference on the Applications of Evolutionary Computation, 2017, pp. 471–486.

[45] K. Price, R. M. Storn, J. A. Lampinen, Differential evolution: a practical approach to global optimization, Springer Science & Business Media, 2006.

[46] A. Liaw, M. Wiener, Classification and regression by randomforest, R news 2 (2002) 18–22.

[47] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: an update, ACM SIGKDD explorations newsletter 11 (2009) 10–18.

[48] V. Sugumaran, V. Muralidharan, K. Ramachandran, Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing, Mechanical systems and signal processing 21 (2007) 930–942.

[49] M. C. de Souto, I. G. Costa, D. S. de Araujo, T. B. Ludermir, A. Schliep, Clustering cancer gene expression data: a comparative study, BMC bioinformatics 9 (2008) 497.

[50] M. C. De Souto, P. A. Jaskowiak, I. G. Costa, Impact of missing data imputation methods on gene expression clustering and classification, BMC bioinformatics 16 (2015) 64.

[51] Q. Yu, Y. Miche, E. Eirola, M. Van Heeswijk, E. SéVerin, A. Lendasse, Regularized extreme learning machine for regression with missing data, Neurocomputing 102 (2013) 45–51.

[52] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine learning research 7 (2006) 1–30.