Qi ZX, Wang HZ, Wang AJ. Impacts of dirty data on classification and clustering models: An experimental evaluation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 36(4): 806–821 July 2021. DOI 10.1007/s11390-021-1344-6

Impacts of Dirty Data on Classification and Clustering Models: An Experimental Evaluation

Zhi-Xin Qi, Hong-Zhi Wang*, *Distinguished Member*, *CCF*, *Member*, *ACM*, *IEEE*, and An-Jie Wang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

E-mail: {qizhx, wangzh}@hit.edu.cn; 1161910522@stu.hit.edu.cn

Received January 31, 2021; accepted June 27, 2021.

Abstract Data quality issues have attracted widespread attentions due to the negative impacts of dirty data on data mining and machine learning results. The relationship between data quality and the accuracy of results could be applied on the selection of the appropriate model with the consideration of data quality and the determination of the data share to clean. However, rare research has focused on exploring such relationship. Motivated by this, this paper conducts an experimental comparison for the effects of missing, inconsistent, and conflicting data on classification and clustering models. From the experimental results, we observe that dirty-data impacts are related to the error type, the error rate, and the data size. Based on the findings, we suggest users leverage our proposed metrics, sensibility and data quality inflection point, for model selection and data cleaning.

Keywords data quality, classification, clustering, model selection, data cleaning

1 Introduction

Data quality has become a serious issue which cannot be overlooked in both database and machine learning communities. We call the data with data quality problems as "dirty data". Clearly, for a classification or clustering task, dirty data in both training and test datasets affect the accuracy. Therefore, we have to know the relationship between the quality of input datasets and the accuracy of results. Based on such relationship, we could select an appropriate model with the consideration of data quality issues and determine the share of data to clean.

Due to the large collection of classification and clustering models, it is difficult for users to decide which model should be adopted. The effects of data quality on models are helpful for model selection. Therefore, exploring dirty-data impacts on models is in demand.

Before a classification or clustering task, data cleaning is necessary to guarantee the data quality. Various data cleaning approaches have been proposed, e.g., data repairing with integrity constraints [1, 2], knowledge-based cleaning systems [3, 4], and crowdsourced data cleaning [3, 5]. These methods improve data quality dramatically, but the costs of data cleaning are still expensive [6]. If we know how dirty data affect the accuracy of models, we could clean data selectively according to the accuracy requirements instead of cleaning the entire dirty data. As a result, the data cleaning costs are reduced. Therefore, the study of the relationship between the data quality and the accuracy of results is urgently needed.

Most of the existing researches have been devoted to developing methods of data cleaning and noise reduction [1–9]. Some literatures also have concentrated on the analyses of noise-robust models from the perspective of class labels [10]. Rare work has been conducted to study the impact of attribute noise [11]. Unfortunately, there is no existing research to explore the impacts of dirty data in terms of data quality dimen-

Regular Paper

Special Section on AI4DB and DB4AI

This work was supported by the National Natural Science Foundation of China under Grant Nos. U1866602 and 71773025, the CCF-Huawei Database System Innovation Research Plan under Grant No. CCF-HuaweiDBIR2020007B, and the National Key Research and Development Program of China under Grant No. 2020YFB1006104.

*Corresponding Author

sions. Thus, this paper aims to fill this gap, which brings the following challenges.

1) There are insufficient experiments to compare and analyze the sensibility and tolerability of classification and clustering models comprehensively. This makes it difficult for users to determine which model is the best and decide which to use. Thus, the first challenge is to design proper experimental approaches to find the model which is the least sensitive and the most tolerant to dirty data.

2) Existing measures of classification and clustering models, such as precision, recall, and F-measure, have been proposed, aiming to test the accuracy of a model, while none of them is able to quantify the dirty-data sensibility and tolerability of a model. Therefore, how to define appropriate evaluation metrics is the second challenge.

In the light of these challenges, we select 12 classical data mining and machine learning models, and attempt to explore their sensibility and tolerability to dirty data. To achieve this goal, we generate dirty datasets based on nine typical datasets for classification and clustering with the consideration of various factors, such as data quality dimension, dirty data rate, and data size. Then, we experimentally compare the performance of different models on various kinds of dirty data. Two novel metrics are proposed to measure the dirty-data sensibility and tolerability of a model. Based on the evaluation results, we provide suggestions for model selection and data cleaning.

In summary, our contributions in this paper are listed as follows.

1) To evaluate dirty-data impacts on models, we propose two novel metrics, sensibility and data quality inflection point. These two metrics are used to measure the fluctuation degree and the dirty-data tolerability of a model.

2) Using the proposed metrics, we conduct an experimental comparison for the dirty-data impacts from various data quality dimensions on classification and clustering models. To the best of our knowledge, this is the first paper that studies this issue.

3) Based on the experimental results, we discover some factors which affect the model performance and provide guidelines of model selection and data cleaning for users.

The rest paper is organized as follows. We discuss related work in Section 2. Section 3 introduces our evaluation methodology. Our experimental results and analyses are presented in Section 4. We discuss the lessons learned from the evaluation and provide guidelines of model selection and data cleaning in Section 5. Section 6 concludes our paper.

2 Related Work

In this section, we overview the related work and analyze our differences. The related work is divided into the following classes.

Data Cleaning and Noise Reduction. A great amount of work has been devoted to cleaning the training or the testing data to reduce noise. In database community, various data cleaning methods have been presented in terms of different data quality dimensions. For example, Wang *et al.* proposed a hybrid human-machine approach to identify the same entity in datasets [5]. Beskales *et al.* discussed simultaneous modifications of the inconsistent data and the functional dependencies [1]. Chu *et al.* combined heterogeneous rules and proposed a holistic method for the automatic repair of dirty data [2]. Chu *et al.* built a data cleaning system powered by knowledge bases and crowdsourcing [3]. Hao *et al.* proposed detective rules to detect and repair wrong relational data [4]. These approaches improve data quality dramatically, but the costs of data cleaning are expensive since they repair the entire dirty data. Instead of proposing a data cleaning method, we study the relationship between the quality of datasets and the accuracy of models. The relationship provides guidelines for selective data cleaning and the cleaning costs are reduced.

In machine learning community, it is important to reduce the consequences of noise. Gamberger and Lavrač first tested a series of noise detection and elimination algorithms in data preprocessing for inductive learning models and suggested that a relaxed consensus saturation filter is a very good solution in noise reduction [7]. García-Laencina *et al.* analyzed the missing data problem in pattern classification tasks [8]. They compared some methods used for handling missing values, and provided solutions based on the experimental results. Lim developed an automatic correction algorithm to reduce noisy city names [9]. Instead of proposing a noise reduction method, our paper focuses on exploring the relationship between the noise and the learning models. According to the relationship and given data, we could select an appropriate machine learning model which achieves a high accuracy.

Impacts of Noise. Several researches have focused on the impacts of noise on data mining and machine

learning. Song and Zhang [12] studied an interesting problem of clustering and repairing over dirty data at the same time. They formalized it as an integer linear programming (ILP) problem and proposed an LP solution without calling a solver. Upon the solution, they designed an approximation algorithm which improves the accuracy of both clustering and cleaning. The work by Zhu and Wu is the most closely related to our work [11]. They investigated the relationship between attribute noise and classification accuracy, the impacts of different attribute noise, and possible solutions in handling attribute noise. In the light of this work, we attempt to explore the impacts of dirty data in terms of data completeness, consistencies, and identity. To achieve this goal, we experimentally evaluate the relationship between the dirty data and the model accuracy on different types of dirty data, including missing, inconsistent, and conflicting data. According to the evaluation, we provide guidelines of model selection and data cleaning for users.

Analyses of Noise-Robust Models. Many studies have concentrated on the analyses of which data mining and machine learning models are more robust to noise. For example, Frénay and Verleysen discussed the potential consequences of the label noise on classification, and suggested some noise-robust and noise-tolerant models [10]. However, existing studies differentiated noise into class noise and attribute noise, and analyzed class-noise-robust models for classification or regression tasks. Instead, our paper divides dirty data into missing values, inconsistent values, and conflicting values from the perspective of data quality dimensions. We then test the classification and clustering models on dirty data to explore dirty-data impacts on models. Based on the relationship of model accuracy and dirty data, we recommend dirty-data-robust models for data mining and machine learning tasks.

3 Evaluation Methodology

In this section, we introduce our experimental methodology, including datasets (Subsection 3.1), classification and clustering models (Subsection 3.1), setup (Subsection 3.1), data quality dimensions (Subsection 3.2), and evaluation measures (Subsection 3.3).

3.1 Datasets, Models, and Setup

We selected nine typical datasets from UCI public datasets① with various types and sizes. Their basic information is shown in Table 1. Due to the completeness and correctness of these original datasets, we injected errors of different rates from different data quality dimensions into them, and generated different kinds of dirty datasets. Then, we compared the performance of various models on them. In experimental evaluation, the original datasets were used as the baseline, and the accuracy of models was measured based on the results on original datasets.

Table 1. Datasets Information

Name	Number of Attributes	Number of Records
Iris	4	150
Ecoli	8	336
Car	6	1 728
Chess	36	3 196
Adult	14	48 842
Seeds	7	210
Abalone	8	4 177
HTRU	9	17 898
Activity	3	67 651

We selected 14 classical classification and clustering models. Their types and parameter settings are shown in Table 2. We chose these models since they are always used as competitive models [13–16].

All experiments were conducted on a machine powered by two Intel® Xeon® E5-2609 v3@1.90 GHz CPUs and 32 GB memory, under CentOS7. All the models were implemented in C++ and compiled with g++ 4.8.5.

3.2 Dimensions of Data Quality

Data quality has many dimensions [17]. For each dimension, there is a corresponding dirty data type. In the research field, dirty data are classified into a variety of types. Most existing researches focus on improving data quality for missing data, inconsistent data, and conflicting data [18, 19]. Thus, in this paper, we focus on these three basic types. For these types, the corresponding data quality dimensions are completeness, consistency, and entity identity.

Missing data refer to the values that are missing from databases. For example, in Table 3, the values of $t1$[Country] and $t2$[City] are missing data.

①http://archive.ics.uci.edu/ml/datasets.html, June 2021.

Table 2. Models Information

Name	Type	Parameter Setting
Decision Tree	Classification	$purity = 0.9$
K-Nearest Neighbor	Classification	$k = 600$, $distance\_type =$ euclidean
Naive Bayes	Classification	–
Bayesian Network	Classification	–
Logistic Regression	Classification	$\alpha = 0.001$, $max\_iteration = 7$
Random Forests	Classification	$max\_height = 3$, $train\_times = 100$, $max\_container = 40$
XGBoost	Classification	$learning\_rate = 0.1$, $max\_depth = 10$, $random\_state = 30$
Multi-Layer Perceptron	Classification	$n\_hidden = 1\,024$, $epoch\_num = 100$, $learning\_rate = 0.1$
K-Means	Clustering	$distance\_type =$ euclidean
LVQ	Clustering	$learning\_rate = 0.2$, $max\_iteration = 1\,000\,000$, $distance\_type =$ euclidean
CLARANS	Clustering	$max\_times = 30$
DBSCAN	Clustering	$\lambda = 2$, $min\_pts = 4$, $max\_cluster = 100$
BIRCH	Clustering	$max\_cf = 20$, $max\_radius = 30\,000$
CURE	Clustering	$\alpha = 0.4$, $point\_num = 10$

Table 3. Student Information

	Student No.	Name	City	Country
$t1$	170302	Alice	NYC	
$t2$	170302	Steven		FR
$t3$	170304	Bob	NYC	U.S.A
$t4$	170304	Bob	LA	U.S.A

Inconsistent data are identified as the violations of functional dependencies which describe the semantic constraints of data. For example, a functional dependency "[Student No.] $\rightarrow$ [Name]" in Table 3 means that Student No. determines Name. As Table 3 shows, $t1$[Student No.] $= t2$[Student No.], but $t1$[Name] $\neq t2$[Name]. Thus, the values of $t1$[Student No.], $t1$[Name], and $t2$[Name] are inconsistent.

Conflicting data refer to different values which describe an attribute of the same entity. For example, in Table 3, both $t3$ and $t4$ describe Bob's information, but $t3$[City] and $t4$[City] are different. Thus, $t3$[City] and $t4$[City] are conflicting data.

3.3 Evaluation Metrics

Since there are class labels in the selected original datasets for classification and clustering, we used standard precision, recall, and F-measure to evaluate the effectiveness of classification and clustering models. These measures are computed as follows.

$$Precision = \frac{\sum_{i=1}^{n_c} \frac{rc_i}{rn_i}}{n_c},$$

where rc_i is the number of records which are correctly classified or clustered to class i, rn_i is the number of records which are classified or clustered to class i, and n_c is the number of classes.

$$Recall = \frac{\sum_{i=1}^{n_c} \frac{rc_i}{r_i}}{n_c},$$

where rc_i is the number of records which are correctly classified or clustered to class i, r_i is the number of records of class i, and n_c is the number of classes.

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

However, these metrics only show us the variations of accuracy. They were not possible to measure the fluctuation degrees quantitatively. Therefore, we propose novel metrics to evaluate dirty-data impacts on models. The first metric sensibility is defined as follows.

Definition 1. *Given the values y_a, y_{a+x}, ..., y_{a+bx} of measure y of a model with the error rate of given data $a\%$, $(a+x)\%, (a+2x)\%, ..., (a+bx)\%$ $(a \geqslant 0, x > 0, b > 0)$, respectively, the sensibility of a model on dirty data is computed as $|y_a - y_{a+x}| + |y_{a+x} - y_{a+2x}| + ... + |y_{a+(b-1)x} - y_{a+bx}|$.*

Note that in this paper, error rate denotes the proportion of dirty values in the given data, a determines the starting point of the error rate, x denotes the step size of the error rate, and b represents the number of the error rates used in the experimental evaluation minus one.

Sensibility aims to measure the fluctuation degree of a model to dirty data. The larger the value of sensibility, the larger the fluctuation degree. Accordingly, dirty data have a greater impact on the model. Therefore, sensibility is able to evaluate the degree of dirty-data impact on a model. Here, we explain the computation of sensibility with Fig.1 as an example.

Example 1. Since the values of precision (P) of the Decision Tree model with the missing rate of given data

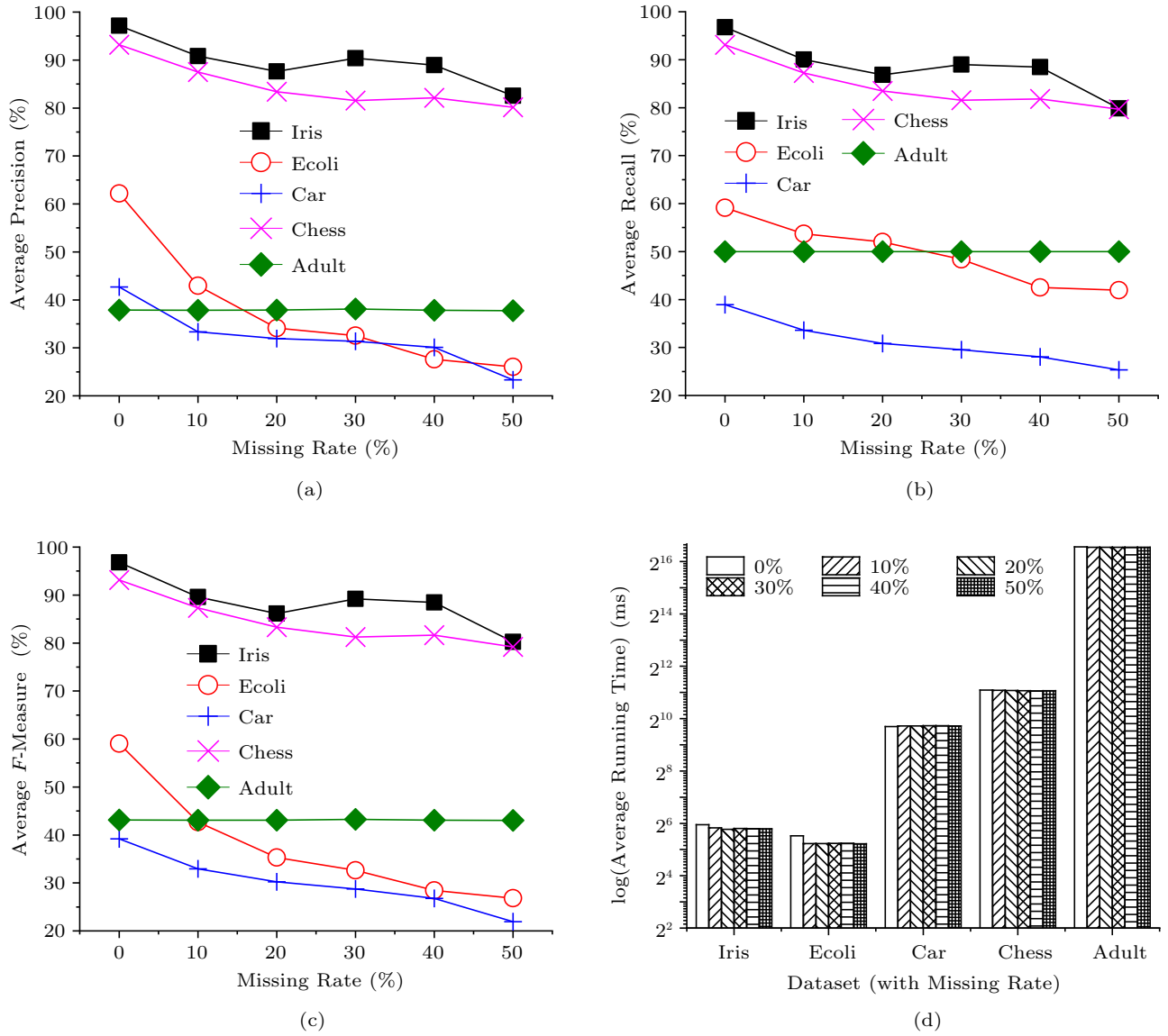

Fig.1. Results of k-nearest neighbor: varying missing rate. (a) Precision. (b) Recall. (c) F-measure. (d) Running time.

0%, 10%, ..., 50% are given, we compute the sensibility for each dataset as shown in Table 4. From Table 4, we obtain the average sensibility of Decision Tree is 25.89%.

Though sensibility measures the sensibility of a model, we could not determine the error rate at which a model is unacceptable. Motivated by this, we define the second novel metric data quality inflection point as follows.

Definition 2. *Given the values* $y_a, y_{a+x}, ..., y_{a+bx}$ *of measure* y *of a model with the error rate of given data* $a\%, (a+x)\%, (a+2x)\%, ..., (a+bx)\%$ $(a \geqslant 0, x >$

Table 4. Sensibility Computation

Dataset	Sensibility
Iris	$\lvert 78.37\% - 84.16\% \rvert + \lvert 84.16\% - 78.08\% \rvert + \lvert 78.08\% - 74.36\% \rvert + \lvert 74.36\% - 64.99\% \rvert + \lvert 64.99\% - 58.71\% \rvert = 31.24\%$
Ecoli	$\lvert 63.47\% - 62.93\% \rvert + \lvert 62.93\% - 53.97\% \rvert + \lvert 53.97\% - 50.93\% \rvert + \lvert 50.93\% - 48.07\% \rvert + \lvert 48.07\% - 34.50\% \rvert = 28.97\%$
Car	$\lvert 81.33\% - 60.93\% \rvert + \lvert 60.93\% - 43.70\% \rvert + \lvert 43.70\% - 42.87\% \rvert + \lvert 42.87\% - 40.47\% \rvert + \lvert 40.47\% - 35.47\% \rvert = 45.86\%$
Chess	$\lvert 82.17\% - 78.17\% \rvert + \lvert 78.17\% - 76.53\% \rvert + \lvert 76.53\% - 75.77\% \rvert + \lvert 75.77\% - 75.90\% \rvert + \lvert 75.90\% - 75.57\% \rvert = 6.86\%$
Adult	$\lvert 80.50\% - 75.27\% \rvert + \lvert 75.27\% - 71.30\% \rvert + \lvert 71.30\% - 72.93\% \rvert + \lvert 72.93\% - 71.53\% \rvert + \lvert 71.53\% - 67.23\% \rvert = 16.53\%$

$0, b > 0$), *respectively, model accuracy* M, *and a number* k ($k > 0$), *if* $M_1 \geqslant M_2$ *when* $y_1 \geqslant y_2$, *and* $y_{a\%} - y_{(a+ix)\%} > k$ ($0 < i \leqslant b$), *data quality inflection point* (*DQIP for brief*) *is* $\min\{(a + (i-1)x)\%\}$. *If* $y_{a\%} - y_{(a+bx)\%} \leqslant k$, *DQIP is* $\min\{(a + bx)\%\}$. *If* $M_1 \leqslant M_2$ *when* $y_1 \geqslant y_2$, *and* $y_{(a+ix)\%} - y_{a\%} > k$ ($0 < i \leqslant b$), *DQIP is* $\min\{(a + (i-1)x)\%\}$. *If* $y_{(a+bx)\%} - y_{a\%} \leqslant k$, *DQIP is* $\min\{(a + bx)\%\}$.

Note that k denotes the acceptable decreasing degree. DQIP is defined to measure the error rate at which a model is acceptable. The larger the value of DQIP, the larger the error rate at which a model is acceptable and accordingly, the higher the error-tolerability of a model. Therefore, DQIP is useful to evaluate the error-tolerability of a model. Here, we take Fig.1 as an example to explain DQIP of a model.

Example 2. We know the values of Precision of the Decision Tree model with the missing rate of given data 0%, 10%, ..., 50%, and set k as 10%. As shown in Table 5, we compute DQIP for each dataset and obtain the average DQIP of Decision Tree is 28%.

Table 5. DQIP Computation

Dataset	DQIP
Iris	Because $y_{0\%} - y_{40\%} = 78.37\% - 64.99\% = 13.38\%$ $> 10\%$, DQIP $= 40\% - 10\% = 30\%$
Ecoli	Because $y_{0\%} - y_{30\%} = 63.47\% - 50.93\% = 12.54\%$ $> 10\%$, DQIP $= 30\% - 10\% = 20\%$
Car	Because $y_{0\%} - y_{10\%} = 81.33\% - 60.93\% = 20.40\%$ $> 10\%$, DQIP $= 10\% - 10\% = 0\%$
Chess	Because $y_{0\%} - y_{50\%} = 82.17\% - 75.57\% = 6.60\%$ $\leqslant 10\%$, DQIP $= 50\%$
Adult	Because $y_{0\%} - y_{50\%} = 80.50\% - 67.23\% = 13.27\%$ $> 10\%$, DQIP $= 50\% - 10\% = 40\%$

4 Evaluation Results and Analyses

We experimentally study the impacts of dirty data on the selected 14 models and analyze the experimental results. In this paper, we show some representative results in this paper and the other results are available online[②].

4.1 Results and Analyses of Classification Models

4.1.1 Varying Missing Rate

To evaluate the impacts of missing data on classification models, we delete values from original datasets randomly and generated five datasets whose missing rate is 10%, 20%, 30%, 40%, and 50%, respectively. For each tuple, we randomly chose one or more attributes and deleted the corresponding values. We used 10-fold cross validation, and generated the training data and the testing data randomly. In the training and testing process, we imputed numerical missing values with the average values and captured categorical ones with the maximal values. The experimental results of KNN are depicted in Fig.1.

Based on the results, we have the following observations. First, for well-performed models (precision, recall, or F-measure is larger than 80% on the original datasets), as the data size increases, precision, recall, or F-measure of models becomes less sensitive, except Logistic Regression. The reason is that if the data size is large, the amount of clean data is large enough to reduce the impacts of missing data. However, Logistic Regression needs to set parameters in its regression function and the parameter computation is easily affected by missing data. Thus, when the data size rises, the amount of missing data becomes larger, which has a larger impact on Logistic Regression.

Second, in Table 6, we obtain the sensibility orders of different classification models on precision, recall, and F-measure. For instance, the sensibility order on precision is "Bayesian Network > Logistic Regression > Naive Bayes > Decision Tree > Random Forests > XGBoost > Multi-Layer Perceptron > KNN". Thus, the least sensitive model is KNN. This is because that as the missing rate rises, the increasing missing values may not affect k nearest neighbors. Even if k nearest neighbors are affected, they are not necessarily voted for the final class label. In addition, the most sensitive model is Bayesian Network. The reason is that the increasing missing data could affect the computation of posterior probabilities, which would directly impact the classification results.

Third, in Table 7, we obtain the DQIP orders of different classification models on precision, recall, and F-measure. For example, the DQIP order on precision is "Decision Tree > Naive Bayes = Random Forests > KNN = XGBoost > Logistic Regression = Multi-Layer Perceptron > Bayesian Network". Therefore, for precision and F-measure, the most incompleteness-tolerant model is Decision Tree. This is because that decision tree models only use splitting features for classification. As the missing rate rises, the increasing missing data may not affect splitting features. For recall, the most incompleteness-tolerant model is Random Forests. This

②https://github.com/qizhixinhit/Dirty-dataImpacts/blob/master/impacts%20of%20dirty%20data.pdf, June 2021.

Table 6. Sensibility Results of Classification and Clustering Models

Model	Missing (%)			Inconsistent (%)			Conflicting (%)		
	Prec.	Rec.	F-Measure	Prec.	Rec.	F-Measure	Prec.	Rec.	F-Measure
DT	25.89	31.11	26.64	35.41	40.94	38.33	16.09	21.56	16.45
KNN	**18.09**	**13.18**	**17.45**	**21.84**	**19.21**	**20.93**	**11.39**	**6.70**	**9.32**
NB	27.04	23.37	26.40	29.48	37.18	35.49	15.10	21.85	20.33
BN	46.40	34.04	35.37	33.29	21.53	23.15	17.26	15.18	16.01
LR	38.26	18.73	30.69	37.84	28.10	38.83	31.74	18.51	25.60
RF	25.77	24.57	29.39	39.21	34.86	40.74	27.93	15.85	27.53
XGBoost	24.47	25.90	25.81	31.64	33.42	32.76	12.27	12.64	10.52
MLP	22.81	18.14	18.57	28.94	22.92	25.24	17.70	9.22	11.54
KM	31.06	27.80	32.08	31.83	32.21	35.63	23.79	21.86	25.17
LVQ	**11.94**	**21.14**	19.61	20.55	18.83	21.41	**9.20**	19.57	20.13
CLARANS	34.26	40.16	39.48	31.11	29.45	31.56	20.67	22.64	24.04
DBSCAN	15.89	22.88	**17.16**	**20.40**	**10.39**	**12.34**	18.64	**9.55**	**16.10**
BIRCH	32.58	44.56	32.90	24.32	22.48	19.40	15.16	22.44	16.52
CURE	38.68	32.71	39.23	28.81	32.90	32.67	32.74	29.11	32.62

Note: Prec.: precision; Rec.: recall.

Table 7. DQIP Results of Classification and Clustering Models ($k = 10\%$)

Model	Missing (%)			Inconsistent (%)			Conflicting (%)		
	Prec.	Rec.	F-Measure	Prec.	Rec.	F-Measure	Prec.	Rec.	F-Measure
DT	**28**	26	**28**	18	16	16	**50**	**50**	**50**
KNN	24	32	20	22	**22**	**22**	40	50	40
NB	26	28	24	22	12	12	50	40	40
BN	20	26	24	16	26	26	46	50	50
LR	22	28	16	16	14	16	32	34	32
RF	26	**50**	10	**26**	14	8	42	38	34
XGBoost	24	26	22	16	14	12	48	48	48
MLP	22	28	22	18	16	16	46	48	48
KM	38	32	32	28	22	22	44	38	38
LVQ	**44**	**40**	**48**	28	14	20	44	44	**40**
CLARANS	2	2	0	22	18	18	34	34	28
DBSCAN	30	40	30	**32**	**44**	**34**	36	**50**	36
BIRCH	24	20	24	20	26	26	**50**	34	38
CURE	18	18	16	20	18	16	32	34	24

Note: Prec.: precision; Rec.: recall.

is because the increasing missing values may not affect splitting attributes. Even if the splitting attributes are affected, the impact of missing data will be reduced due to multiple classifiers. For precision and recall, the least incompleteness-tolerant model is Bayesian Network. This is because the increasing missing data would change the posterior probabilities, which could affect the classification results directly. For F-measure, the least incompleteness-tolerant model is Random Forests. The reason is that F-measure on original datasets (error rate is 0%) is high. When few missing values exist in the datasets, F-measure drops a lot.

Fourth, the results of precision, recall, and F-measure of Bayesian Network on Ecoli when the missing rate is 0% are much slower than those when the missing rate is 10%. Also, the precision value of Random Forests on Iris when the missing rate is 10% is much higher than that when the missing rate is 0%. This is because that the data size of Ecoli or Iris is relatively small, which easily causes overfitting of models. Although better models are trained with higher data quality, the model performance on testing data may be worse. This observation further confirms that it is unnecessary to clean the entire dirty data.

Fifth, as the data size increases, the running time of classification models rises more with the increasing missing rate. This is because as the data size rises, the amount of missing data becomes larger, which introduces more uncertainty to algorithms. Accordingly, the uncertainty of running time increases.

4.1.2 *Varying Inconsistent Rate*

To evaluate the impact of inconsistency on classification models, we injected inconsistent values to the original datasets randomly varying inconsistent rate and generated five datasets whose inconsistent rate is 10%, 20%, 30%, 40%, and 50%, respectively. First, we randomly selected a certain number of tuples. For each selected tuple, we constructed a corresponding tuple with an inconsistent value according to the functional dependency. Then, we inserted all the new tuples into the given data. In this way, we generated inconsistent data with controllable repairability [20]. We used 10-fold cross validation, and generated the training data and the testing data randomly. Since inconsistent data make no difference to the training and testing process, we trained and tested models on the generated inconsistent data. The experimental results of KNN are depicted in Fig.2.

Based on the results, we have the following observations. First, in Table 6, we obtain the sensibility orders of classification models on precision, recall, and F-measure. Thus, the least sensitive model is KNN. The reason is similar to that of the least sensitive model varying missing rate. For precision and F-measure, the most sensitive model is Random Forests. And for recall, the most sensitive model is Decision Tree. These are caused by the fact that as the inconsistent rate increases, more and more incorrect values cover the correct ones in training models, which leads to inaccurate classification results. Since the base classifiers in Random Forests are decision trees, the reason for Random

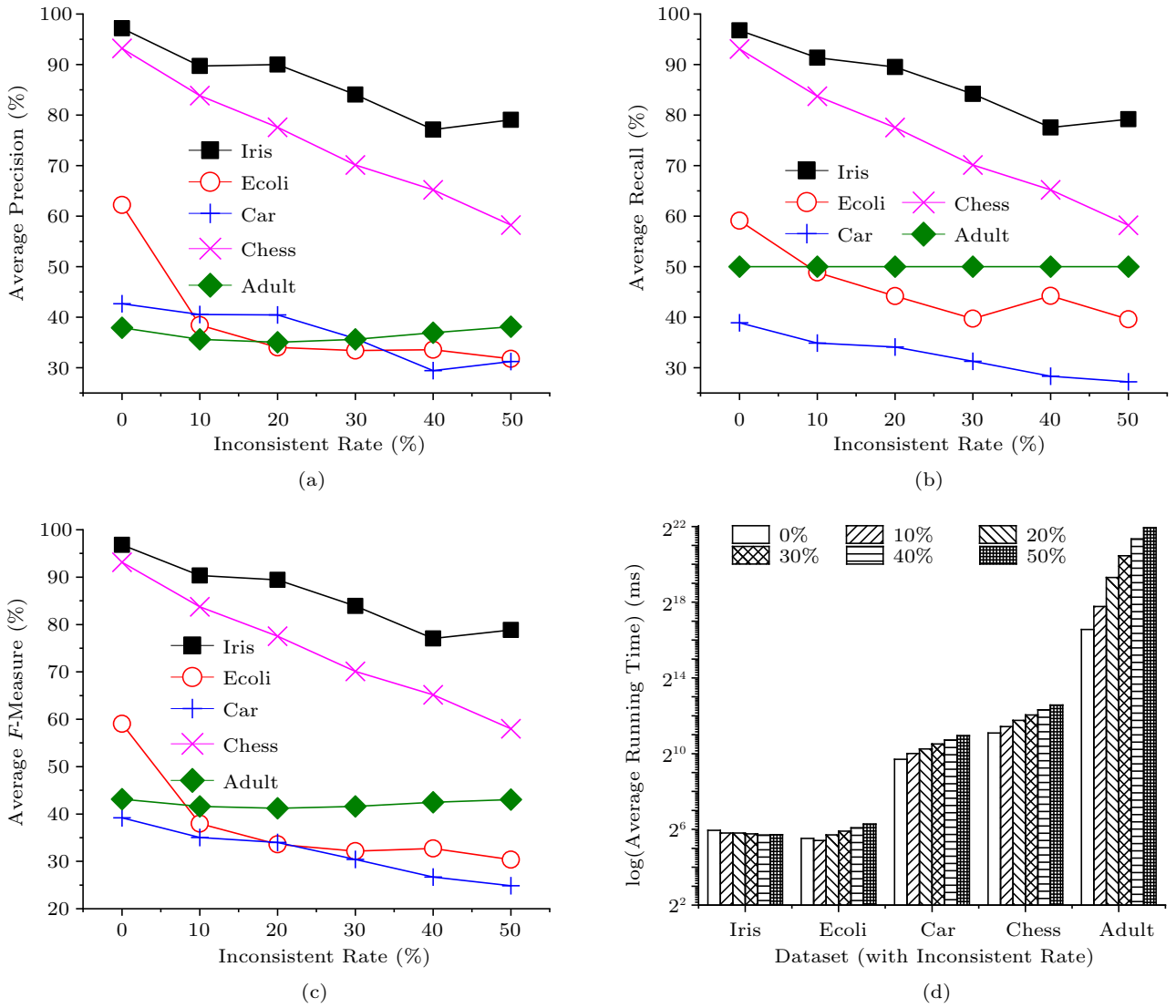

Fig.2. Results of k-nearest neighbor: varying inconsistent rate. (a) Precision. (b) Recall. (c) F-measure. (d) Running time.

Forests is the same as that for Decision Tree.

Second, in Table 7, we obtain the DQIP orders of classification models on precision, recall, and F-measure. Therefore, for precision, the most inconsistency-tolerant model is Random Forests. The reason is similar to that of the most incompleteness-tolerant model varying missing rate. For recall and F-measure, the most inconsistency-tolerant model is Bayesian Network. This is because inconsistent values contain incorrect ones and correct ones. Hence, the incorrect values have little effect on the computation of posterior probabilities. Accordingly, the classification results may not be affected. For precision, the least inconsistency-tolerant models are Bayesian Network, Logistic Regression, and XGBoost. For recall, the least inconsistency-tolerant model is Naive Bayes. And for F-measure, the least inconsistency-tolerant model is Random Forests. These are because precision, recall, and F-measure of these models on original datasets (error rate is 0%) are high. When few inconsistent values are injected, precision, recall, and F-measure drop dramatically.

Third, the observation of running time varying inconsistent rate is the same as that when the missing rate is varied.

4.1.3 *Varying Conflicting Rate*

To evaluate the impacts of conflicting data on classification models, we injected conflicting values to original datasets randomly and generated five datasets whose conflicting rate is 10%, 20%, 30%, 40%, and 50%, respectively. First, we randomly selected a certain number of tuples. For each tuple, we constructed a corresponding tuple with one attribute value modified. Then, we inserted the new tuples into the given data. We used 10-fold cross validation, and generated the training data and the testing data randomly. Since conflicting data makes no difference to the training and testing process, we trained and tested models on the generated conflicting data. The experimental results of KNN are depicted in Fig.3.

First, the observation of the relationship between data size and model sensibility varying conflicting rate is the same as that when the missing rate is varied.

Second, in Table 6, we obtain the sensibility orders of classification models on precision, recall, and F-measure. Thus, the least sensitive model is KNN. The reason is similar to that of the least sensitive model varying the missing rate of given data. For precision, the most sensitive model is Logistic Regression. This is because the parameter computation of the regression function is easily affected by the increasing conflicting values, which causes an inaccurate logistic regression model. For recall, the most sensitive model is Naive Bayes. This is because the incorrect values in the increasing conflicting data affect the computation of posterior probabilities in the Bayes theorem. For F-measure, the most sensitive model is Random Forests. The reason is the same as that of the most sensitive model when the inconsistent rate is varied.

Third, in Table 7, we obtain the DQIP orders of classification models on precision, recall, and F-measure. Therefore, the most conflict-tolerant model is Decision Tree. The reason is similar to that of the most incompleteness-tolerant model. The least conflict-tolerant model is Logistic Regression. This is because the incorrect values in conflicting data affect the parameter computation of logistic regression models. Accordingly, the classification accuracy drops dramatically.

Fourth, the results of precision, recall, and F-measure of Decision Tree, XGBoost, and Multi-Layer Perceptron on Ecoli when the missing rate is 0% are much lower than those when the missing rate is 10%. This is because the data size of Ecoli is relatively small, which easily causes overfitting of models. This observation further confirms that it is unnecessary to clean the entire dirty data.

Fifth, the observation of running time varying conflicting rate is the same as that when the missing rate is varied.

4.2 Results and Analyses of Clustering Models

4.2.1 *Varying Missing Rate*

To evaluate missing-data impacts on clustering models, we deleted values from original datasets randomly and generated five datasets whose missing rate is 10%, 20%, 30%, 40%, and 50%, respectively. For each tuple, we randomly chose one or more attributes and deleted the corresponding values. In the clustering process, we imputed numerical missing values with the average values, and captured categorical ones with the maximal values. The experimental results of LVQ are depicted in Fig.4.

Based on the results, we have the following observations. First, in Table 6, we obtain the sensibility orders of clustering models on precision, recall, and F-measure. Thus, for precision and recall, the least sensitive model is LVQ. This is because LVQ is a super-

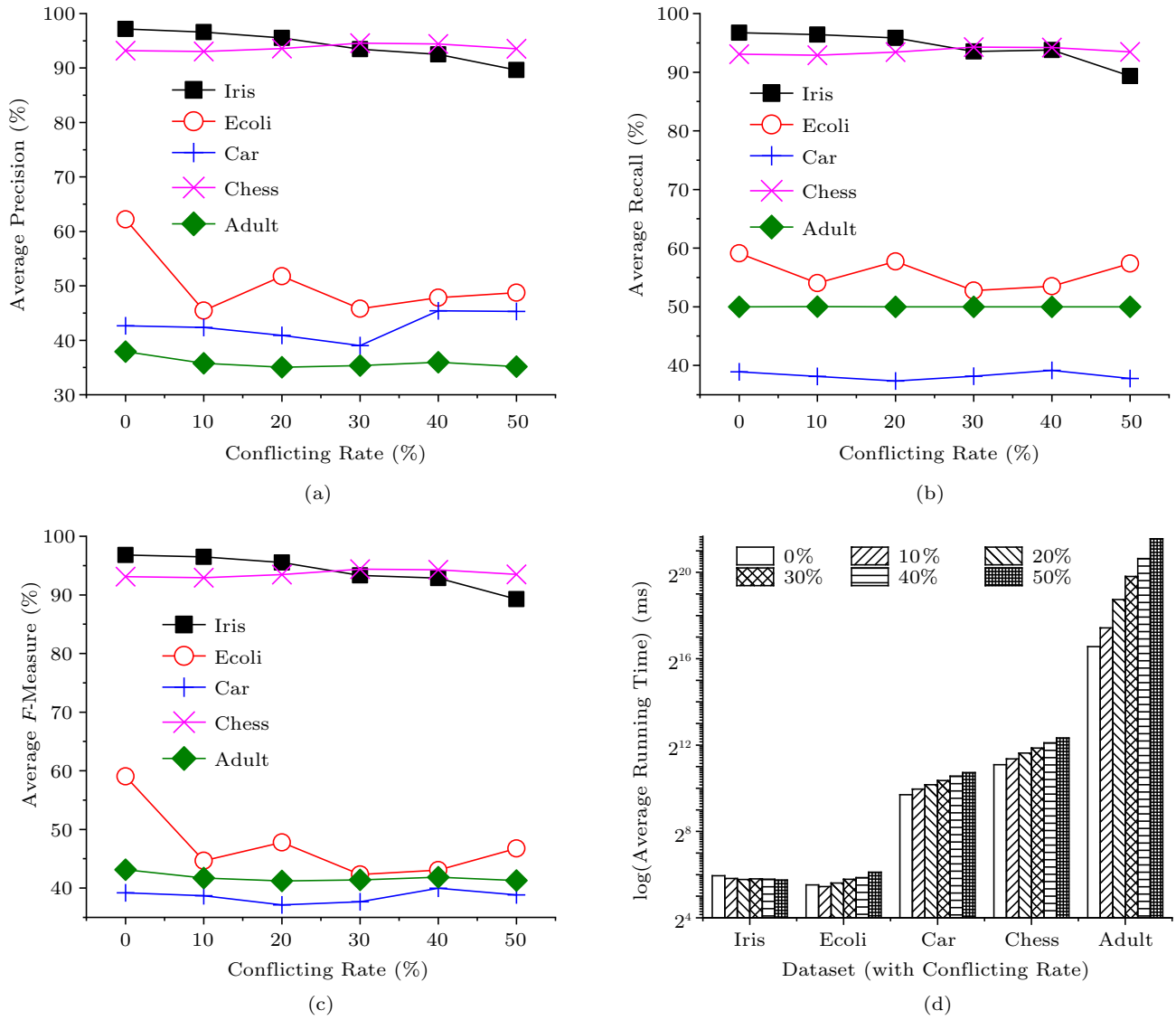

Fig.3. Results of k-nearest neighbor: varying conflicting rate. (a) Precision. (b) Recall. (c) F-measure. (d) Running time.

vised clustering model on the basis of marked labels. Hence, there is little chance to be affected by missing values. For F-measure, the least sensitive model is DBSCAN. This is due to the fact that DBSCAN eliminates all noise points at the beginning of the model, which makes it more resistant to missing values. For precision, the most sensitive model is CURE. This is because the location of representative points in CURE is easily effected by missing values, which causes inaccurate clustering results. For recall, the most sensitive model is BIRCH. This is due to the fact that missing data could impact the construction of clustering feature tree in BIRCH, which directly leads to wrong clustering results. For F-measure, the most sensitive model is CLARANS. This is because that the computation of cost difference in CLARANS is susceptible to missing values, which makes some points clustered incorrectly.

Second, in Table 7, we obtain the DQIP orders of clustering models on precision, recall, and F-measure. Therefore, the most incompleteness-tolerant model is LVQ. This is because LVQ is a supervised clustering model based on marked labels. Hence, there is little chance for it to be affected by missing values. The least incompleteness-tolerant model is CLARANS. This is due to the fact that the computation of cost difference in CLARANS is susceptible to missing data, which causes inaccurate clustering results.

Third, the results of precision, recall, and F-measure of K-Means on Abalone when the missing rate is 0% are much lower than those when the missing rate

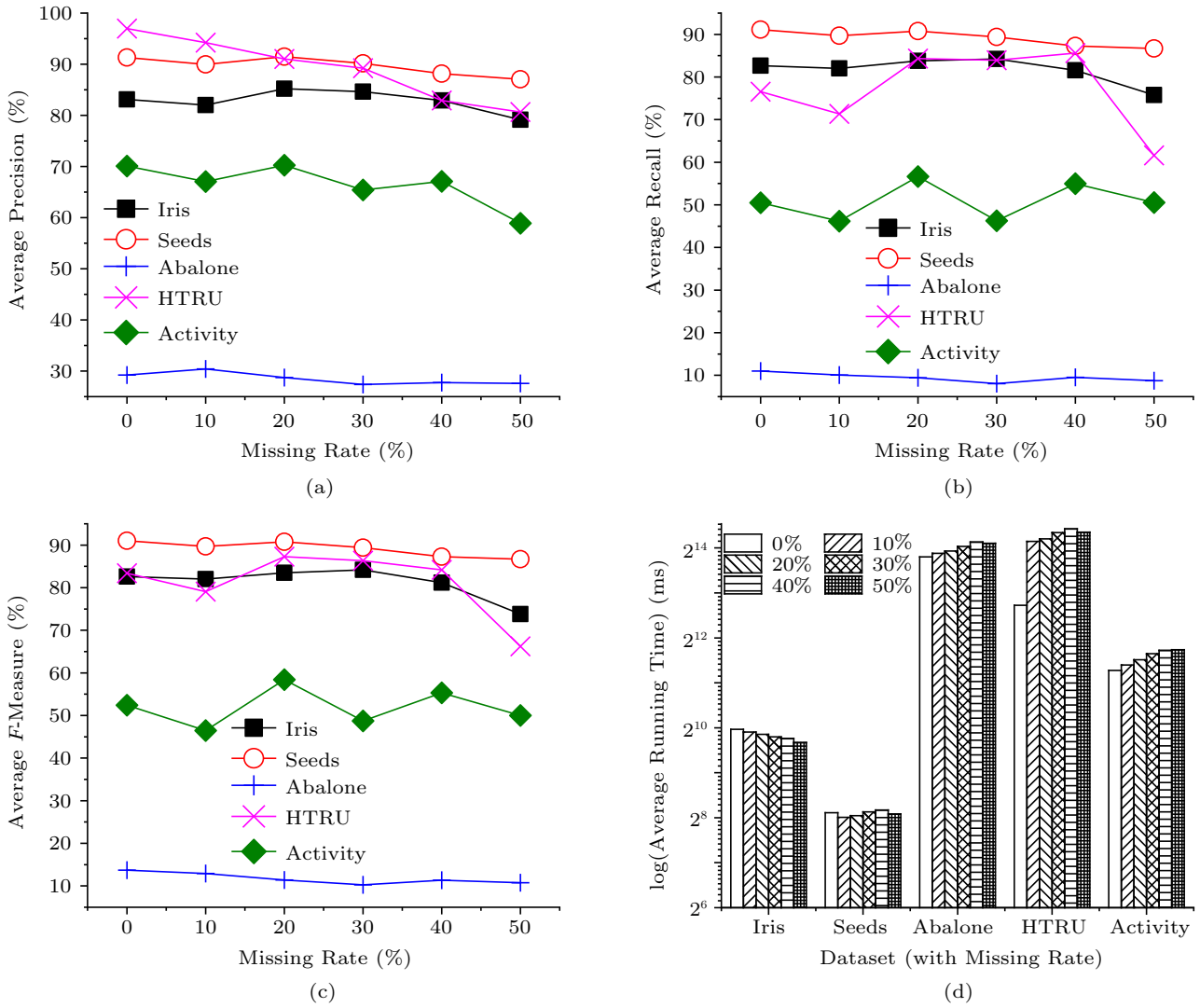

Fig.4. Results of LVQ: varying missing rate. (a) Precision. (b) Recall. (c) F-measure. (d) Running time.

is 10%. The recall and F-measure results of BIRCH on HTRU with the missing rate of 10% are much higher than those with that of 0%. Also, the results of precision, recall, and F-measure of CURE on Activity when the missing rate is 0% are much lower than those when the missing rate is 10%. The phenomenon shows fewer missing values may cause better clustering results. This observation further confirms it is unnecessary to clean the entire dirty data.

Fourth, as the data size increases, the running time of clustering models rises more with the increasing missing rate. This is because as the data size rises, the amount of missing data becomes larger, which introduces more uncertainty to models. Accordingly, the uncertainty of running time increases.

4.2.2 *Varying Inconsistent Rate*

To evaluate inconsistent-data impacts on clustering models, we injected inconsistent values to original datasets randomly and generated five datasets whose inconsistent rate is 10%, 20%, 30%, 40%, and 50%, respectively. First, we randomly selected a certain number of tuples. For each selected tuple, we constructed a corresponding tuple with an inconsistent value according to the functional dependency. Then, we inserted all the new tuples into the given data. In this way, we generated inconsistent data with controllable repairability [20]. Since inconsistent data make no difference to the clustering process, we trained clustering models on the generated inconsistent data. The experimental results of DBSCAN are depicted in Fig.5.

Based on the results, we have the following obser-

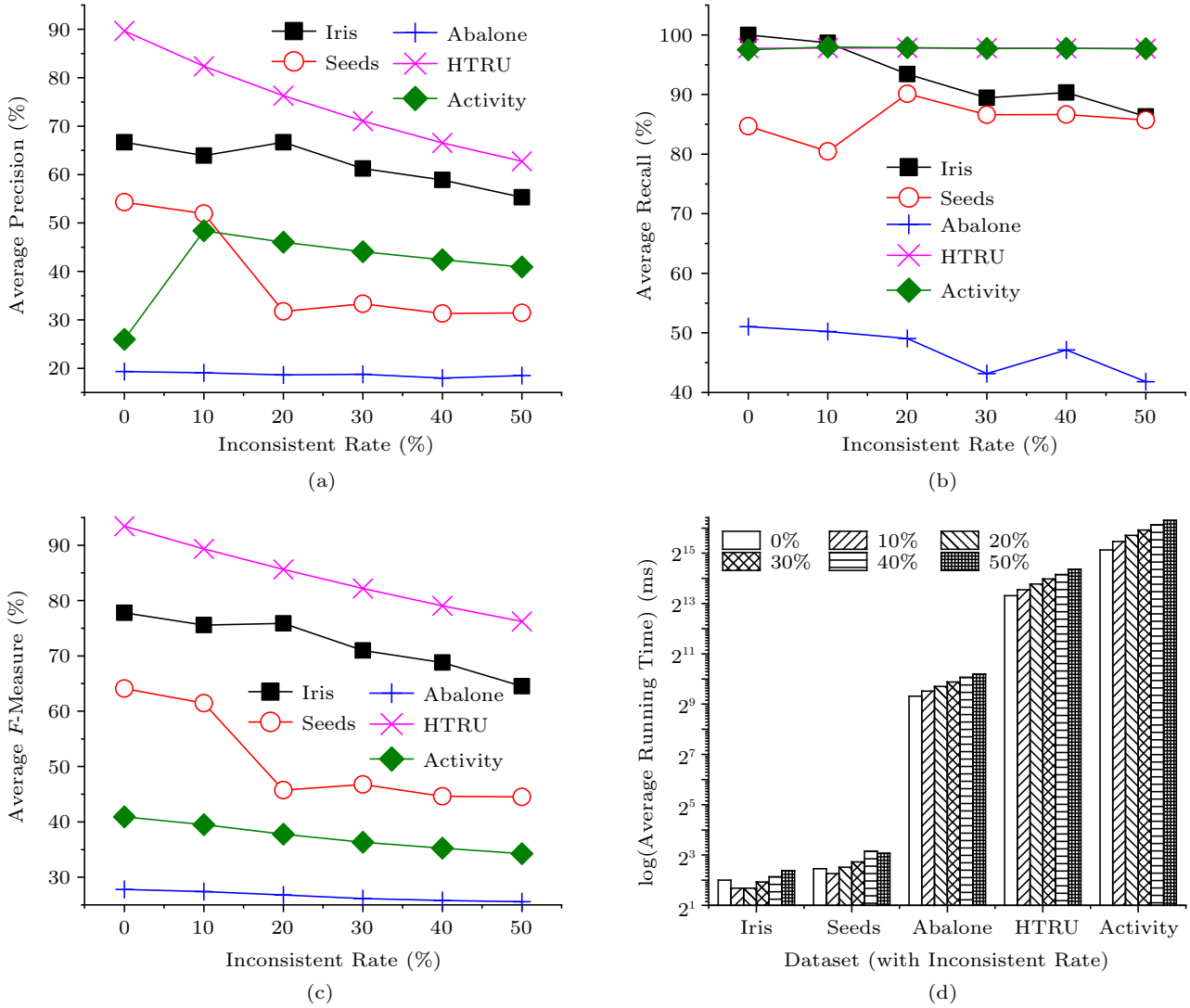

Fig.5. Results of DBSCAN: varying inconsistent rate. (a) Precision. (b) Recall. (c) F-measure. (d) Running time.

vations. First, for well-performed models (precision, recall, or F-measure is larger than 80% on original datasets), as the data size increases, precision, recall, or F-measure of models fluctuates more widely, except DBSCAN. This is because the amount of inconsistent values becomes larger as the data size rises. The increasing incorrect data have more effects on the clustering process. However, DBSCAN discards noise points at the beginning of the model. When the data size rises, the number of correct values becomes larger. Accordingly, the proportion of eliminated points reduces and the impact on DBSCAN decreases.

Second, in Table 6, we obtain the sensibility orders of clustering models on precision, recall, and F-measure. Thus, the least sensitive model is DBSCAN. The reason is similar to that of the least sensitive model varying missing rate. For precision and F-measure, the most sensitive model is K-Means. This is due to the fact that the computation of centroids is susceptible to incorrect values, which causes wrong clustering results. For recall, the most sensitive model is CURE. The reason is similar to that of the most sensitive model varying missing rate.

Third, in Table 7, we obtain the DQIP orders of clustering models on precision, recall, and F-measure. Therefore, the most inconsistency-tolerant model is DBSCAN. This is because DBSCAN eliminates all noise points at the beginning of the model, which makes it more resistant to inconsistent data. For precision, the least inconsistency-tolerant models are BIRCH and CURE. For recall, the least inconsistency-tolerant model is LVQ. For F-measure, the least inconsistency-

tolerant model is CURE. These are due to the fact that the distance computation of these models is susceptible to incorrect values, which causes inaccurate clustering results.

Fourth, the results of precision, recall, and F-measure of K-Means on Abalone when the inconsistent rate is 0% are much lower than those when the inconsistent rate is 10%. The precision result of DBSCAN on Activity with the inconsistent rate of 10% is much higher than those with the inconsistent rate of 0%. Also, the results of precision, recall, and F-measure of CLARANS and CURE on Activity when the inconsistent rate is 0% are much lower than those when the inconsistent rate is 10%. The phenomenon shows fewer inconsistent values may cause better clustering results. This observation further confirms it is unnecessary to clean the entire dirty data.

Fifth, the observation of running time varying inconsistent rate is the same as that when the missing rate is varied.

4.2.3 *Varying Conflicting Rate*

To evaluate the impacts of conflicting data on clustering models, we injected conflicting values to original datasets randomly and generated five datasets whose conflicting rate is 10%, 20%, 30%, 40%, and 50%, respectively. First, we randomly selected a certain number of tuples. For each tuple, we constructed a corresponding tuple with one attribute value modified. Then, we inserted the new tuples into the given data. Since conflicting data makes no difference to the clustering process, we trained clustering models on the generated conflicting data. The experimental results of DBSCAN are depicted in Fig.6.

Based on the results, we have the following observations. First, in Table 6, we obtain the sensibility orders of clustering models on precision, recall, and F-measure. Thus, for precision, the least sensitive model is LVQ. The reason is similar to that of the least sensitive model varying the missing rate of given data. For recall and F-measure, the least sensitive model is DBSCAN. The reason has been discussed in Subsection 4.2.1. The most sensitive model is CURE. The reason is similar to that of the most sensitive model varying missing rate.

Second, in Table 7, we obtain the DQIP orders of clustering models on precision, recall, and F-measure. Therefore, for precision, the most conflict-tolerant model is BIRCH. This is because the conflicting data contain correct ones and incorrect ones, which makes the construction of clustering feature tree insusceptible to incorrect values. For recall, the most conflict-tolerant model is DBSCAN. The reason is similar to that of the most inconsistency-tolerant model varying the inconsistent rate of given data. For F-measure, the most conflict-tolerant model is LVQ. The reason is similar to that of the most incompleteness-tolerant model varying missing rate. The least conflict-tolerant model is CURE. This is due to the fact that the location of representative points in CURE is easily affected by conflicting values, which makes data points clustered inaccurately.

Third, the results of precision, recall, and F-measure of K-Means on Abalone when the conflicting rate is 0% are much lower than those when the conflicting rate is 10%. Also, the precision, recall, and F-measure results of CLARANS and CURE on Activity with the conflicting rate of 0% are much lower than those when the conflicting rate is 10%. The phenomenon shows fewer conflicting values may cause better clustering results. This observation further confirms it is unnecessary to clean the entire dirty data.

Fourth, the observation of running time varying conflicting rate is the same as that when the missing rate is varied.

5 Lessons Learned

In this section, we first discuss the lessons learned from the evaluation. Based on the discussions, we provide guidelines of model selection and data cleaning for users. Also, we give suggestions for future work to researchers and practitioners.

5.1 Lessons Learned from Evaluation on Classification Models

According to the evaluation results and analyses of classification models, we have the following findings.

• Dirty-data impacts are related to the error type and the error rate. Thus, it is necessary to detect the error rate of each error type in the given data.

• For the models whose precision, recall, or F-measure is larger than 80% on original datasets, as the data size rises, precision, recall, or F-measure of the models becomes less sensitive, except Logistic Regression. Since the parameter k in DQIP was set as 10%, the candidate models of which precision, recall, or F-measure is larger than 70% are acceptable.

• As the data size increases, precision, recall, and F-measure of Logistic Regression become more sensitive.

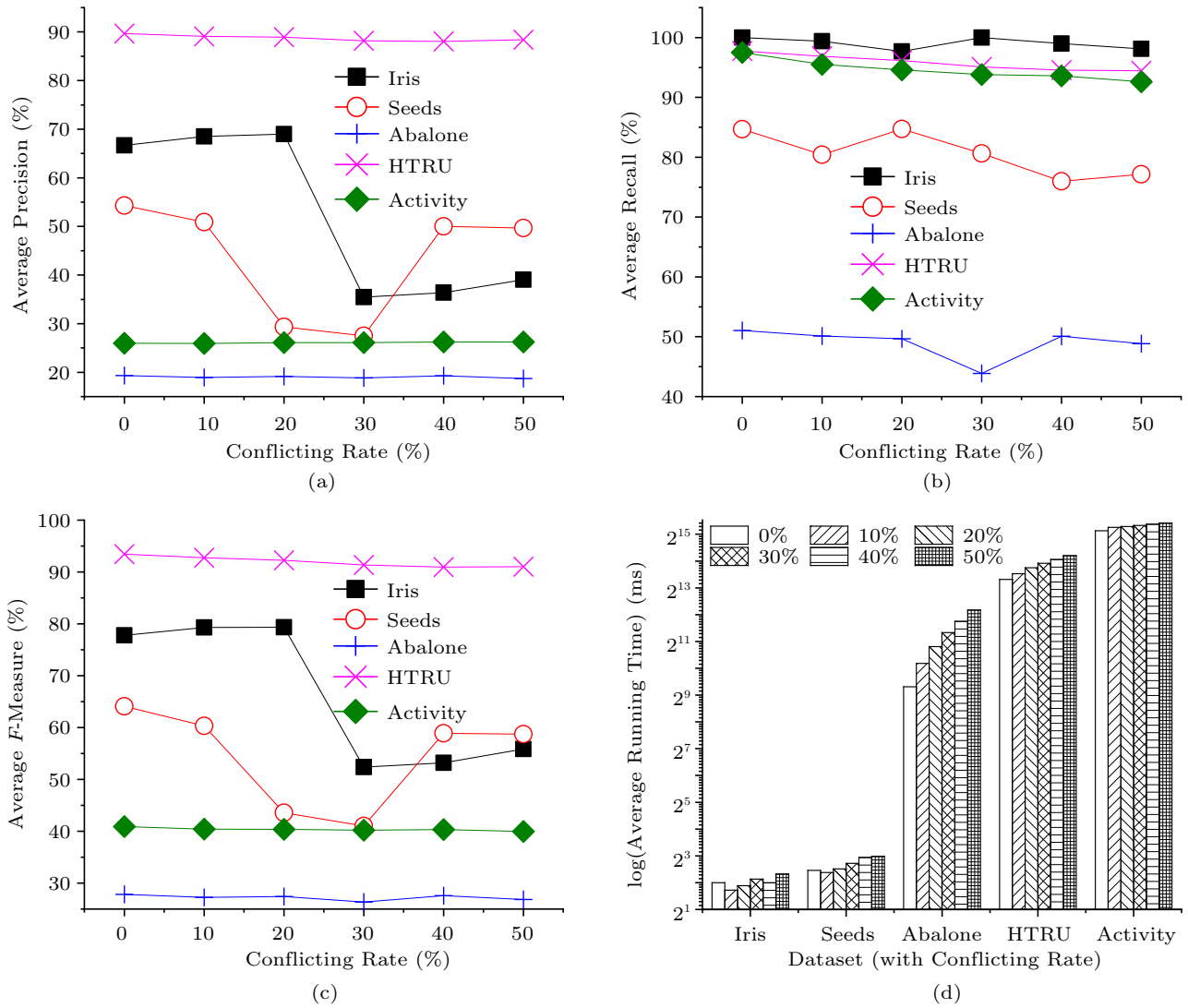

Fig.6. Results of DBSCAN: varying conflicting rate. (a) Precision. (b) Recall. (c) F-measure. (d) Running time.

• It is unnecessary to clean the entire dirty data before a classification task.

• Since precision, recall, or F-measure of the selected classification model becomes unacceptable when the error rate is above its corresponding DQIP, the error rate of each dirty data type needs to be cleaned below the value of its DQIP.

• As the data size increases, the running time of classification models rises more with the increasing error rate.

5.2 Guidelines of Classification Model Selection and Data Cleaning

Based on the discussion, we suggest users select classification model and clean dirty data according to the following steps.

1) Users are suggested to detect the error rates (e.g., missing rate, inconsistent rate, conflicting rate) of the given data [2, 3].

2) According to the given task requirements (e.g., well performance on precision, recall, or F-measure), we suggest users select the candidate models of which precision, recall, or F-measure on the given data is better than 70%.

3) Logistic Regression is not recommended if the given data size is larger than 100 M.

4) According to task requirements and the error type which takes the largest proportion, we suggest users leverage our proposed experimental methodology to obtain the corresponding sensibility order and choose the least sensitive classification model.

5) According to the selected model, task require-

ments, and error rates of the given data, we suggest users use our evaluation methodology to obtain the corresponding DQIP orders and clean each type of dirty data until the error rate is below its DQIP.

5.3 Lessons Learned from Evaluation on Clustering Models

According to the evaluation results and analyses of clustering models, we have the following findings.

• Dirty-data impacts are related to the error type and the error rate. Thus, it is necessary to detect the error rate of each error type in the given data.

• For models whose precision, recall, or F-measure is larger than 80% on original datasets, as the data size rises, precision, recall, or F-measure of the models becomes more sensitive, except DBSCAN. Since the parameter k in DQIP was set as 10%, the candidate models of which precision, recall, or F-measure is larger than 70% are acceptable.

• As the data size increases, precision, recall, or F-measure of DBSCAN becomes less sensitive.

• It is unnecessary to clean the entire dirty data before a clustering task.

• Since precision, recall, or F-measure of the selected clustering model becomes unacceptable when the error rate is above its corresponding DQIP, the error rate of each dirty data type needs to be cleaned below the value of its DQIP.

• As the data size increases, the running time of clustering models rises more with the increasing error rate.

5.4 Guidelines of Clustering Model Selection and Data Cleaning

According to the discussions, we suggest users select clustering model and clean dirty data according to the following steps.

1) Users are suggested to detect the error rates (e.g., missing rate, inconsistent rate, conflicting rate) of the given data [2, 3].

2) According to the given task requirements (e.g., well performance on precision, recall, or F-measure), we suggest users select the candidate models of which precision, recall, or F-measure on the given data is better than 70%.

3) DBSCAN is not recommended if the given data size is smaller than 10 M.

4) According to task requirements and the error type which takes the largest proportion, we suggest users leverage our proposed experimental methodology to obtain the corresponding sensibility order and choose the least sensitive clustering model.

5) According to the selected model, task requirements, and error rates of the given data, we suggest users use our evaluation methodology to obtain the corresponding DQIP orders and clean each type of dirty data until the error rate is below its DQIP.

5.5 Suggestions for Future Work

This work opens many noteworthy avenues for future work, which are listed as follows.

• Since dirty-data impacts on classification and clustering models are valuable, their effects on other kinds of models (e.g., association rules mining) need to be tested.

• Dirty-data impacts are related to error type, error rate, data size, and model performance on original datasets. Hence, it is valuable to explore how to construct a model with these parameters to predict dirty-data impacts is in demand.

• Since different users have different requirements of precision, recall, or F-measure, how to clean data in demand needs a solution.

6 Conclusions

In this paper, we conducted a comprehensive experimental evaluation for the impacts of dirty data on classification and clustering models. From the experimental results, we validated that our proposed metrics, sensibility and DQIP, are useful to explore dirty-data impacts on models. In addition, we discovered three factors affecting the model performance that are the error type, the error rate, and the data size. Based on the evaluation findings, we suggested users detect the given datasets before a classification or clustering task. Then, according to the requirements of precision, recall, or F-measure, users are suggested to use our proposed methodology to obtain the orders of sensibility and DQIP. We believe that these two orders are valuable to select appropriate machine learning models and clean dirty data selectivity.

References

[1] Beskales G, Ilyas I F, Golab L, Galiullin A. On the relative trust between inconsistent data and inaccurate constraints. In *Proc. the 29th IEEE Int. Conf. Data Engineering*, Apr. 2013, pp.541-552. DOI: 10.1109/ICDE.2013.6544854.

[2] Chu X, Ilyas I F, Papotti P. Holistic data cleaning: Putting violations into context. In *Proc. the 29th IEEE Int. Conf. Data Engineering*, Apr. 2013, pp.458-469. DOI: 10.1109/ICDE.2013.6544847.

[3] Chu X, Morcos J, Ilyas I F, Ouzzani M, Papotti P, Tang N, Ye Y. KATARA: A data cleaning system powered by knowledge bases and crowdsourcing. In *Proc. the 36th ACM Int. Conf. Management of Data*, May 2015, pp.1247-1261. DOI: 10.1145/2723372.2749431.

[4] Hao S, Tang N, Li G, Li J. Cleaning relations using knowledge bases. In *Proc. the 33rd IEEE Int. Conf. Data Engineering*, Apr. 2017, pp.933-944. DOI: 10.1109/ICDE.2017.141.

[5] Wang J, Kraska T, Franklin M J, Feng J. CrowdER: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 2012, 5(11): 1483-1494. DOI: 10.14778/2350229.2350263.

[6] Dallachiesa M, Ebaid A, Eldawy A, Elmagarmid A, Ilyas I F, Ouzzani M, Tang N. NADEEF: A commodity data cleaning system. In *Proc. the 34th ACM Int. Conf. Management of Data*, Jun. 2013, pp.541-552. DOI: 10.1145/2463676.2465327.

[7] Gamberger D, Lavrač N. Conditions for Occam's razor applicability and noise elimination. In *Proc. the 9th Springer Eur. Conf. Machine Learning*, Apr. 1997, pp.108-123. DOI: 10.1007/3-540-62858-4_76.

[8] García-Laencina P J, Sancho-Gómez J L, Figueiras-Vidal A R. Pattern classification with missing data: A review. *Neural Computing and Applications*, 2010, 19(2): 263-282. DOI: 10.1007/s00521-009-0295-6.

[9] Lim S. Cleansing noisy city names in spatial data mining. In *Proc. the 2010 Int. Conf. Information Science and Applications*, Apr. 2010. DOI: 10.1109/ICISA.2010.5480390.

[10] Frénay B, Verleysen M. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Networks and Learning Systems*, 2013, 25(5): 845-869. DOI: 10.1109/TNNLS.2013.2292894.

[11] Zhu X, Wu X. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 2004, 22(3): 177-210. DOI: 10.1007/s10462-004-0751-8.

[12] Song S, Li C, Zhang X. Turn waste into wealth: On simultaneous clustering and cleaning over dirty data. In *Proc. the 21st ACM Int. Conf. Knowledge Discovery and Data Mining*, Aug. 2015, pp.1115-1124. DOI: 10.1145/2783258.2783317.

[13] Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In *Proc. the 23rd ACM Int. Conf. Machine Learning*, Jun. 2006, pp.161-168. DOI: 10.1145/1143844.1143865.

[14] Caruana R, Karampatziakis N, Yessenalina A. An empirical evaluation of supervised learning in high dimensions. In *Proc. the 25th ACM Int. Conf. Machine Learning*, Jul. 2008, pp.96-103. DOI: 10.1145/1390156.1390169.

[15] Ghotra B, McIntosh S, Hassan A E. Revisiting the impact of classification techniques on the performance of defect prediction models. In *Proc. the 37th IEEE/ACM Int. Conf. Software Engineering*, May 2015, pp.789-800. DOI: 10.1109/ICSE.2015.91.

[16] Kirchner K, Zec J, Delibašić B. Facilitating data preprocessing by a generic framework: A proposal for clustering. *Artificial Intelligence Review*, 2016, 45(3): 271-297. DOI: 10.1007/s10462-015-9446-6.

[17] Sidi F, Panahy P H S, Affendey L S, Jabar M A, Ibrahim H, Mustapha A. Data quality: A survey of data quality dimensions. In *Proc. the 2nd IEEE Int. Conf. Information Retrieval and Knowledge Management*, Mar. 2012, pp.300-304. DOI: 10.1109/InfRKM.2012.6204995.

[18] Fan W, Geerts F. Capturing missing tuples and missing values. In *Proc. the 29th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems*, Jun. 2010, pp.169-178. DOI: 10.1145/1807085.1807109.

[19] Getoor L, Machanavajjhala A. Entity resolution: Theory, practice & open challenges. *Proceedings of the VLDB Endowment*, 2012, 5(12): 2018-2019. DOI: 10.14778/2367502.2367564.

[20] Arocena P C, Glavic B, Mecca G, Miller R J, Papotti P, Santoro D. Messing up with BART: Error generation for evaluating data-cleaning algorithms. *Proceedings of the VLDB Endowment*, 2015, 9(2): 36-47. DOI: 10.14778/2850578.2850579.

Zhi-Xin Qi is a Ph.D. candidate in School of Computer Science and Technology, Harbin Institute of Technology, Harbin. She received her B.S. degree in information security from Harbin Engineering University, Harbin, in 2016, and her M.S. degree in computer technology from Harbin Institute of Technology, Harbin, in 2018. Her research interests include database, graph data management, and knowledge graph.

Hong-Zhi Wang is a professor and Ph.D. supervisor of Harbin Institute of Technology, Harbin. He received his B.S., M.S., and Ph.D. degrees in computer science and technology from Harbin Institute of Technology, Harbin, in 2001, 2003, and 2008 respectively. He is the secretary general of ACM SIGMOD China and a distinguished CCF member. His research fields include big data management and analysis, database systems, knowledge engineering, and data quality.

An-Jie Wang is currently a Master student in School of Computer Science and Technology, Zhejiang University, Hangzhou. He received his B.S. degree in computer science and technology from Harbin Institute of Technology, Harbin, in 2020. His research interests include data quality, big data management, and computer graphics.