

Received December 10, 2020, accepted December 27, 2020, date of publication December 31, 2020, date of current version January 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3048419

LR-BCA: Label Ranking for Bridge Condition Assessment

KAI WANG¹, TONG RUAN¹, AND FAXIANG XIE²

¹School of Information Science and Engineering, East China University of Science and Technology, Shanghai 20037, China

²College of Civil and Transportation Engineering, Hohai University, Nanjing 210098, China

Corresponding author: Tong Ruan (ruantong@ecust.edu.cn)

This work was supported by the National Natural Science Foundation of China for Open Link Data (LOD) Quality of Use Evaluation Theory and Method Research under Grant 61772201.

ABSTRACT Bridge condition assessment (BCA) plays an important role in modern bridge management. Existing assessment methods are time-consuming, labor-intensive and error-prone. The use of machine learning for BCA can effectively solve the above problems. However, the large amount of label noise in the dataset severely affected the performance of the BCA model. In this paper, we present an effective label ranking approach for BCA (LR-BCA). Our proposed LR-BCA method considers the natural order relationship between bridge condition ratings. Moreover, a heuristic data cleaning (HDC) approach is proposed for cleaning bridge condition dataset. The HDC method firstly identifies all the label conflict examples, then iteratively filters out the noise. Experimental results on real-world dataset confirm the effectiveness of the HDC method and demonstrate that our proposed LR-BCA method achieves 99% Top-2 accuracy, which is highly competitive compared to baseline methods.

INDEX TERMS Bridge condition assessment, data cleaning, machine learning, label ranking.

I. INTRODUCTION

Bridges play an important role in societies as they assist the daily movement of people and wares. However, the collapses of bridges happen occasionally because of the deteriorating of bridge components along time. Due to the crucial importance of bridges, some countries used bridge management systems (BMS) and intelligent transportation systems (ITS) to manage and to utilize bridge condition information for traffic management [1]. Bridge management aims to determine the optimal strategy for operating bridges under the consideration of adequate security, thereby minimizing operational costs during the bridge's life cycle. Therefore, BCA is an important part of BMS and ITS. It is possible to provide decision support for bridge management through BCA.

Table 1 illustrates the attributes of bridge condition data. Generally, the bridge is subdivided into several components in inspection phase. Bridge inspectors observe at site to record bridge components condition data and assess components condition with visual inspection. However, the accuracy of the recorded bridge condition data is not high because of the involvement of inspectors' experience and amounts of fuzzy bridge condition information [2]. Inaccurate recorded

The associate editor coordinating the review of this manuscript and approving it for publication was Li He¹.

TABLE 1. Bridge condition data attributes list. There are 418 defect attributes and each attribute is described as "type of defects+span length+defect property".

Main classes	Attribute name	Value type
Basic bridge information	Last inspection	Year
	Year built	Numerical
	Traffic	Numerical
Defects	Flow direction	Binary
	prestressed	Binary
	Transverse cracks in the beam floor+10m+cracks quantity	Numerical
	Transverse cracks in the beam floor+10m+sum of cracks width	Numerical
	Abutment reinforcement metal corrosion+10m+cracks quantity+6m+max length	Numerical
End event	other 415 attributes	Numerical
	Bridge condition rating	Numerical (Label)

data may make it difficult to evaluate bridge condition correctly. BCA is a data-driven operation, which needs high-quality data. On the one hand, the wrong assessment of bridges can lead to unnecessary maintenance and reinforcement. On the other hand, any delay or erroneous assessment of bridge conditions may lead to higher maintenance costs and significant safety hazards in the future.

We observe that there are numerous noise in the bridge dataset. Based on the reasons above, it is vital to eliminate “incorrectly labelled” data (or noise) in the data cleaning process after the acquisition of bridge condition data. This paper constructs conflict-pairs to determine noise, and then iteratively filters out them based on the values of the number of conflicts ranking. The cleaning method works well in real-world datasets with significant improvements in model performance. Besides, this paper models the BCA problem as a label ranking problem, which is able to effectively evaluate the bridge condition. The main contributions of this paper are summarized as follows.

- We propose an effective label ranking approach for bridge condition assessment (LR-BCA), which models the bridge rating problem as a label ranking task, and it can effectively evaluate bridge condition and improve error-tolerance ability of the model. To handle conflict examples in BCA dataset, a heuristic data cleaning (HDC) approach is also developed.
- Based on a real-world bridge condition dataset, we experimentally evaluate our LR-BCA approach and HDC approach. Extensive experimental results show that our LR-BCA approach is highly competitive compared to baseline methods. Also, our proposed HDC approach can effectively reduce conflict examples in BCA dataset, thus improving assessment performance.

The rest of the paper is organized as follows. After a brief literature review on existing BCA methods and cleaning methods in Section II, we present a heuristic data cleaning method based on conflict-pairs in Section III. Section IV introduces a label ranking method for bridge condition assessment. Section V provides a set of experiments, as well as the experimental results and analysis. Finally, concluding remarks and potential research directions are given in Section VI.

II. RELATED WORK

A. DATA CLEANING METHODS

The noise in this paper refers to the data with incorrect class labels, i.e. label noise. Data cleaning techniques aims to detect and repair the noise in datasets. Existing methods can be divided into three categories, and we address them as label-noise-robust, datasets-cleaning, and threshold-based filtering approaches, respectively.

1) LABEL-NOISE-ROBUST APPROACHES

The goal of these methods was to modify existing algorithms such that they are less sensitive to noise. An example was C4.5 algorithm [3], which reduced the complexity of the decision tree by pruning strategies. It decreased the probability of decision tree being over-fitting. However, C4.5 algorithm achieved poor performance on datasets with large numbers of attributes. Fürnkranz and Widmer [4] proposed an incremental reduced error pruning (IREP) algorithm in 1994, which could effectively deal with large noise datasets, but its

classification accuracy was lower than C4.5 algorithm. Subsequently, Cohen [5] proposed a repeated incremental pruning to produce error reduction (RIPPER) algorithm, which was an improved IREP algorithm. Compared with C4.5, RIPPER was not only competitive in classification accuracy but also more robust to noise. Nettleton *et al.* [6] experimentally compared the robustness of several algorithms for handling label noise. Experimental results showed that naive Bayes was the most robust algorithm, and support vector machine (SVM) [7] could be easily altered by noise.

2) DATASET-CLEANING APPROACHES

These methods aimed to eliminate noise from training set and train classifiers with “clean” data. These approaches, also known as noise filters [8], were based on the idea that the learning algorithms would benefit from the separation of noise detection and model learning. A standard approach was to use a classification algorithm as the ‘filter’ in noise detection and to use another algorithm as the ‘learner’ in the learning phase. Owing to the fact that k-nearest neighbors (KNN) is highly sensitive to label noise, there were many KNN-based filters where KNN was used to detect noise [9]. Ensemble filters were known to detect noise by the voting result of several different classifiers. Brodley and Friedl [10] proved that detecting noise by the voting results of different classifiers are more effective than collecting noise detection result from a single classifier. Consequently, they proposed an ensemble filter based on C4.5, 1-NN, and linear machine. Similarly, some researchers [11], [12] proposed iterative partition filters that iteratively removed detected noise. In these methods, N filters were learned based on N group of $N - 1$ partitions and voted to detect noise on the entire dataset. García-Gil *et al.* [13] proposed an homogeneous ensemble method (HME) and an heterogeneous ensemble method (HTE) to deal with noise in big data. HME was based on a partition scheme and implemented random forest as a single classifier. HTE, inspired by [10], used random forest, a linear model, and KNN as base classifiers. Consensus Voting Scheme (CVS) and Majority Voting Scheme (MVS) were two classical ensemble settings [14]. The CVS setting is known to achieve a high precision, but it tends to missing a lot of noisy instances. However, The MVS setting often yields high recall, but typically falsely classify numerous clean instances as noise. Samami *et al.* [15] proposed to use a mixed scheme, namely High Agreement Voting Filtering to compensate for the drawbacks of the two settings. They removed the detected strong and semi-strong noise, and then relabeled the weak noise instead of deleting them.

3) THRESHOLD-BASED FILTERING APPROACHES

This category of methods tried to find a way to measure the estimated noise level of examples. For example, Sluban *et al.* [16] used random forest algorithm to identify noise. The agreement levels of decision trees in the forest were used for noise detection. The more the trees had the

same predictions for an example, the more likely it is a clean example. Garcia *et al.* [17] decomposed the multi-class problems into binary sub-problems, and binary filter was used for each sub-problem to calculate the confidence of examples. The confidence levels of examples which were predicted by binary filters were combined. They were used to remove the examples whose confidence levels were higher than the threshold. The threshold-based filters were believed to allow the practitioner to control the removal level. Inspired by the above methods, Sáez *et al.* [8] proposed an iterative noise filter (INFFC) based on the combination of classifiers and noise sensitivity control. The method mainly utilized the ability of k -nearest neighbors based metric approaches with a variety of thresholds. Zerhari *et al.* [18] modified INFFC by using partitioning strategies and removing portion of good examples in each iteration. Sabzevari *et al.* [19] proposed a two stages noise detection method based on the sampling rate of bootstrapping and the degree of disagreement between the individual ensemble members. After determining the optimal sampling rate and the optimal threshold of disagreement, an ensemble method was constructed afresh. Nematzadeh *et al.* [20] proposed to adopt an ensemble filter for noisy instances detection. They computed the average euclidean distance between the suspected noise detected by the ensemble filter and the K -nearest neighbors in the clean set. Finally, they removed and relabeled the suspected noise based on the distance scores. Guan *et al.* [21] introduced to utilize feature selection for better noise filtering performance. They adopt MVS to obtain a noise score for each instance, and then generated noise candidates based on the noise scores. The noise scores and noise candidates were used as the target values and the input instances respectively in the feature selection phase.

The first type of approach depends on the particular modifications of the learning algorithm, and the modification varied from one algorithm to another. The second type often consists of sub-classifiers. They can be either trained on the same dataset using different algorithms, or initialized on different subset of the training set using the same algorithm. As a result, the samples that are indistinguishable to the filter are labeled as noise and removed from the dataset. In other words, the filter only removes the samples that does not match the classification patterns of the sub-classifiers. Problems may rise when the samples are mislabeled as noise when they are hard to distinguish by the filter. As for the third type, a key feature is the removal level, which is subjectively set to select noise sample. This type of method is similar with the ones using noise filter. Therefore, some researchers also refer it as threshold based filter.

To improve the performance of classifiers, different type of methods can often be used together in practical applications. For example, one may use a filter for cleaning in the preprocessing stage, and then use the adapted algorithm in the learning phase. Although noise filters could achieve good performances in some datasets, there are also deficiencies. In real-world scenarios, cleaning through filters may remove

a large amount of normal data, resulting severe information loss when building analytical models. Owing to the same reason, filter related methods did not perform well in bridge condition datasets.

B. BRIDGE CONDITION ASSESSMENT METHODS

Bridge condition assessment (BCA) has attracted enormous attentions from both academia and industry. Various methods have been proposed in the past few decades, and they can be roughly divided into two categories:

Domain knowledge based approaches. This type of method evaluated bridge condition using criteria and indicators based on expert knowledge. For example, Mitsuru and Sinha [22] used Delphi study to develop unified guidelines of bridge condition assessment for inspectors. However, their discussion was limited to bridge deck, i.e. a component of bridge. Kushida and Miyamoto [23] proposed an effective method for correlating empirical knowledge with membership functions based on fuzzy theory to express expert knowledge of bridge condition assessment. Wang and Elhag [24] proposed an evidential reasoning (ER) method for bridge condition assessment to model uncertainties in human subjective assessment. Lately, Dabous and Al-Khayyat [25] introduced a method of combining Monte Carlo simulation with ER, which used pairwise comparison to determine the weights of bridge component condition ratings in whole bridge condition. Alsharqawi *et al.* [26] introduced the extended quality function deployment theory into bridge condition assessment based on visual inspection and ground penetrating radar data. Björnsson *et al.* [27] considered to use the Bayesian decision theory for decisions of bridge condition assessment based on three domain knowledge, namely, modelling sophistication, consideration of uncertainty, and knowledge content. In addition, analytic hierarchy process [28] was also widely used in bridge condition assessment. In the type of methods, a bridge was divided into multiple components to evaluate the whole condition.

Soft computing based approaches. Soft computing technology mainly includes fuzzy logic, genetic algorithm and artificial neural network. These methods were able to directly process the input data without the need to build logical structures. Some researchers [29], [30] introduced the artificial neural network for the bridge condition assessment using the dataset selected from the National Bridge Inventory database. Liu and Zhang [29] considered three primary components as input features of the Convolutional Neural Network, while Nguyen and Dinh [30] utilized Forward Neural Network as the model with eight input features. Flintsch and Chen [31] summarized the applications of the three most common soft computing techniques in infrastructure management, namely artificial neural networks, simulation systems and genetic algorithms. Liu *et al.* [32] used fuzzy c -mean clustering algorithm to evaluate the condition of bridge superstructure, and particle swarm optimization was used to optimize the algorithm. Yusuf and Hamid [33] used artificial neural network and multiple regression analysis to model limited data,

respectively. By comparing and analyzing the performance of two models under various conditions, it concluded that the artificial neural network was more suitable for the bridge condition assessment. Lu *et al.* [34] proposed an ordinal logistic regression algorithm for bridge condition assessment. The algorithm had the advantage of handling the ordinal nature of bridge condition ratings. Li and Burgueño [35] developed four models (i.e. multi-layer perceptron, support vector machine, supervised self-organizing machine and radial basis function network) for the assessment of damage in bridge abutment. Martinez *et al.* [36] studied the performance of five types of predictive models in bridge condition assessment, and concluded that decision tree outperformed the other models including linear regression model, KNN, and neural networks.

Compared to soft computing based approaches, domain knowledge based approaches generally required more expert knowledge. However, expert knowledge was usually very expensive. The soft computing based approaches required enough bridge condition data to learn an adequate model. Although, with the passage of time, a large number of bridge condition data has been accumulated. Due to the subjective factors and the errors in the observations of bridges, there was plenty of incorrect data in the existing dataset, which severely affected the performances of bridge condition assessment models.

III. A HEURISTIC APPROACH FOR DATA CLEANING

Generally, structure of bridge is described by its components, and the main structure includes a series of basic components, such as the main deck, hanging beam, and girder. Owing to defects and damages such as rust, cracks, and deformation, the components condition deteriorated over time. The rating of bridge condition can be determined through the severity of defects and damages. In this section, we introduce the features of bridge condition data, and then illustrate the general form of the heuristic approach proposed in this paper.

A. THE ELEMENT OF BRIDGE CONDITION DATA

An example of data that is used for bridge condition assessment has numbers of attributes and a discrete label.

The attributes include basic bridge information (e.g. year of completion, traffic per data, and whether prestressed bridge or not) and defects (e.g. rust, crack, and deformation). The label is the rating of bridge condition. For the k -th sample in the dataset, the relation between attributes and label can be formulated as equation (1), where $X_1^k, X_2^k, \dots, X_{M_1}^k$ are defects attributes, $X_{M_1+1}^k, X_{M_1+2}^k, \dots, X_{M_1+M_2}^k$ are basic bridge information, and R^k is the condition rating.

$$R^k = f(X_1^k, X_2^k, \dots, X_{M_1}^k, X_{M_1+1}^k, \dots, X_{M_1+M_2}^k) \quad (1)$$

Symbolically, dataset of bridge condition can be organized in Table 2. Without basic bridge information attributes, the bridge condition data can be regarded as monotonic ordinal data.

TABLE 2. The structure of bridge condition dataset.

Example	Defect attributes				Condition rating
1	X_1^1	X_2^1	\dots	$X_{M_1+M_2}^1$	R^1
2	X_1^2	X_2^2	\dots	$X_{M_1+M_2}^2$	R^2
\dots	\dots	\dots	\dots	\dots	\dots
N	X_1^N	X_2^N	\dots	$X_{M_1+M_2}^N$	R^N

In conventional assessment, human inspectors divide the entire bridge into several main components, and combine the assessment results of components to compute the rating. A main component of bridge contains a number of minor structures, and human inspectors assess the main component based on all minor structures. Each main component condition is categorized into five levels, each of which indicates a distinct level of defects and damages. However, discrete condition levels tend to cause wrong assessment. For instance, the main beam transverse cracks that are opened as slight by one human inspector may not be slight for another.

It is difficult to acquire a clean test set because of the complexity of bridge condition assessment. Moreover, a major problem with state-of-the-art cleaning methods is that they always require a clean test set to evaluate its cleaning effectiveness. It makes the state-of-the-art methods inapplicable to the bridge condition assessment, since we cannot easily acquire a clean test set. If we implemented noise filters in test set, we might risk removing the clean data just because it cannot be classified correctly by classifiers. To clean label noise in the dataset, we propose a heuristic data cleaning (HDC) approach, which is model-agnostic.

B. HEURISTIC DATA CLEANING

The HDC includes three main steps, which are detailed in the following subsections, i.e. subsection III-C, III-D, and III-E. Fig. 1 shows the scheme of our approach to filter noisy data. Firstly, the approach does a pairwise comparison in the entire dataset to form a conflict-pairs group. Secondly, the approach counts the number of times that examples appear in the conflict-pairs group and then eliminates noise iteratively based on conflict frequencies. We stop the iteration when the cleaning ratio ≥ 0.3 or the number of conflict examples is equal to zero. Finally, this work uses LR algorithm to evaluate HDC by comparing the performance of the model trained in clean dataset with the one initialized in noisy data.

In the rest of the section, D_1 represents the initial dataset, D_2 represents the dataset at the beginning of the cleaning, and D_3 represents the new dataset after cleaning noisy data. The three main steps are summarized as follows.

- **Constructing conflict-pairs.** It preprocesses the dataset D_1 to obtain D_2 by deleting useless attributes. Considering the examples in D_2 , a conflict-pair is formed when they conform with a given condition by pairwise comparison. All conflict-pairs form a conflict-pairs group.
- **Filtering noise examples.** It counts the frequencies of examples' appearance in the group and constructs a

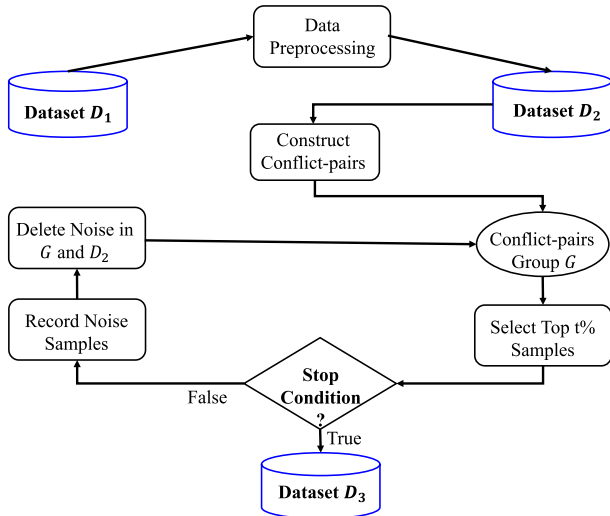


FIGURE 1. Schematic diagram of our proposed HDC method. We stop the iteration when the cleaning ratio ≥ 0.3 or the number of conflict examples is equal to zero.

{Example No. : Frequency} dictionary. Examples in the dictionary are sorted by frequency in descending order, and the top-ranked examples are eliminated iteratively to obtain a new cleaner dataset denoted as D_3 .

- **Training models.** In this step, LR algorithms are used to generate an ordered set of bridge labels to strengthen the models' error-tolerance ability. The LR models M_1 and M_2 are trained in the dataset D_2 and the new dataset D_3 , respectively. The effectiveness of the cleaning method is evaluated by the comparison of two models.

Note that the noise filtered out by HDC are merely suspected label noise. In other words, these identified noise may contain clean examples. In this paper, HDC is based on following important judgements and observations:

- 1) There are pairwise conflicts between bridge examples. According to simple domain knowledge, the more serious the defect level is, the higher the bridge condition rating is. However, for some examples in the dataset, they have higher defects levels, but lower bridge condition ratings. The phenomenon of examples conflicts is very common. As a matter of fact, there are 183, 166 conflict-pairs in our dataset, which accounts for 0.59% of the total example-pairs.
- 2) The attributes of bridge defects are ordinal attributes. Specifically, a bigger defect attribute value indicates a more serious bridge condition, if other attributes hold still.
- 3) The more frequent a bridge example conflicts with others, the more likely it is label noise. For instance, given a bridge example A , if it is conflicted with n examples, the number of conflicts is recorded as n . The bigger n is, the more likely it is label noise.
- 4) Label noise has a negative influence on the performance of models. Apart from decreasing the test

accuracy of algorithms, label noise also prolongs the training time and increases model complexity. The model gains better performance when training in a clean set.

C. CONSTRUCTING CONFLICT-PAIRS

We construct the conflict-pairs group G using initial dataset D_1 . Firstly, preprocessing is conducted on D_1 , including recovering missing values and removing useless columns. The missing values are mostly from basic bridge information attributes. The hot deck method is used to fill the missing basic information attributes with the corresponding values of the nearest bridge sample. The process can be formulated as Formula 2:

$$i_0 \leftarrow \arg \min_i \frac{\sum_{j=j_0}^{M_1+M_2} |e_{i,j} - e_{q,j}|}{M_1 + M_2 - 1} \tag{2}$$

where $e_{i,j}$ is the j -th variable value of i -th example, $e_{q,j}$ is the j -th variable value of the missing example, i_0 is the serial number of the nearest example, M_1 is the number of the defects attributes, and M_2 is the number of the basic bridge information attributes. As for the defects attributes, we merely fill the missing values with zeroes in case the process produces propagating noise.

Then, categorical type of attributes are deleted temporarily to obtain D_2 . Obviously, with the purpose of traversing the entire dataset, HDC is required to compare $\binom{N}{2}$ times where N is the size of dataset. The concept of conflict-pair is defined as follows:

Definition 1: Given two examples A and B , where the rating of A 's condition is R^A and the rating of B 's condition is R^B . Assume that $R^A > R^B$, for all the defects attributes values X_i^A , $i = 1, 2, \dots, M_1$ of A , if the values X_i^B , $i = 1, 2, \dots, M_1$ corresponding to one-to-one in the example B all satisfy $x_i^A \leq x_i^B$, $i = 1, 2, \dots, M$, then (A, B) is a conflict-pair.

Specifically, the construction of conflict-pairs can be formulated as follow:

$$f(A, B) = \begin{cases} \bigcap_{i=1}^{M_1} [(X_i^A - X_i^B) \geq 0] & \text{if } R^A < R^B \\ \bigcap_{i=1}^{M_1} [(X_i^B - X_i^A) \geq 0] & \text{if } R^A > R^B \end{cases} \tag{3}$$

where A and B form a conflict-pair (A, B) , if $f(A, B)$ is true.

To better explain conflict-pairs, a positive example and a counterexample are introduced in Table 3 and 4.

TABLE 3. An example of two examples in a conflict-pair.

Ex.	Defects attributes										Rating
	x_1^k	x_2^k	...	x_{65}^k	x_{66}^k	...	x_{178}^k	x_{179}^k	...	R	
15	3	0.4	0	1	0.1	0	2	30	0		2
720	4	0.6	0	3	0.3	0	3	35	0		1

Example 1: (Positive Example). As shown in Table 3, the example No. 15 has a higher condition rating than the example No. 720, while the defects attributes values of No. 15 are smaller than that of No. 720. This inconsistency between

TABLE 4. An example of two examples not in a conflict-pair.

Ex.	Disease attributes									Rating
	X_1^k	X_2^k	...	X_{43}^k	X_{44}^k	...	X_{148}^k	X_{149}^k	...	
32	3	0.4	0	3	0.1	0	2	30	0	2
65	5	0.6	0	1	0.2	0	4	40	0	1

condition rating and attributes values leads to a conflict. Thus, (15, 720) constitutes a conflict-pair.

Example 2: (Negative Example). As shown in Table 4, No. 32 has a higher condition rating than that of No. 65. Meanwhile, the defect attribute value X_{43} of No. 32 is higher than that of No. 65, while situations with other attribute values are opposite. In this case, No. 32 and No. 65 do not form a conflict-pair.

To detect all conflict-pairs according to Definition 1, we traverse all example-pairs of the dataset. Note that, the construction of conflict-pairs group G ensures that there is no identical conflict-pair.

D. FILTERING OUT NOISE EXAMPLES

In a conflict-pair (A, B) , at least one of the examples has a wrong label according to the Definition 1. Suppose that there are N_k conflict-pairs contain the k -th example. To compute the probability that the label of k -th example is correct, the probability is formulated as follow:

$$P_k = \prod_{i=1}^{N_k} p_{k,i} \quad (4)$$

where $p_{k,i}$ is the probability that the label of k -th example is correct when it belongs to the i -th conflict-pair.

Note that P_k decreases as the increment of N_k . It illustrates that the more times an example appears in conflict-pairs, the more likely it is label noise.

The bottom of Figure 1 illustrates the process of iteratively filtering out label noise. In the i -th iteration, we count the occurrence frequency f_k of the k -th example in G and construct the dictionary $Dict = \{k : f_k\}$, $k = 1, 2, \dots, N$. The examples in $Dict$ are sorted in descending order of frequency f_k . If the stop condition is not satisfied, the top $t\%$ examples are recorded as label noise and then eliminated from D_2 . At the same time, the relevant conflict-pairs in G are eliminated to update G . The next iteration is performed until the stop condition is triggered. New clean dataset D_3 is obtained after the iterations.

Note that the stop condition used in this paper is not unique. In the process of filtering noisy data, users can flexibly set stop conditions according to actual needs.

E. MODEL TRAINING

Since this paper models bridge rating as a LR problem, this work mainly applies LR algorithms. LR algorithms can reveal tendency of examples at bridge condition ratings. Besides, it also increases the error-tolerance abilities of

models. We discuss the effect of label ranking detailedly in Section IV.

To illustrate the effectiveness of HDC, we uses different algorithms and initializes the training on D_2 and D_3 , respectively. Stacking, the integrated learning strategy, is used to learn the model. The stacking algorithm learns data distribution from different data spaces through various base models, and then integrates predictions of the models to obtain the final results. The first layer used in this paper adopts random forest (RF), k-nearest neighbor (KNN), and support vector machine (SVM). The second layer meta-model adopts RF. We use stacking algorithm to prove that HDC can benefit both from single learning and integrated learning algorithms. Furthermore, stacking is a kind of noise-robust algorithm. As discussed in section II.A, different types of noise-dealing methods can often be used together.

Considering the difficulty of obtaining a clean test set, this paper cannot deliberately create some label noise examples to evaluate the proposed cleaning methods, which other papers [37] often do. They compared the determined suspicious noise with the standard set to evaluate the cleaning accuracy. Therefore, in this paper, We utilize the cleaned dataset and the noisy initial dataset to train models M_1 and M_2 , respectively. The performances of M_1 and M_2 are compared to evaluate the effectiveness of HDC.

IV. LABEL RANKING FOR BRIDGE CONDITION ASSESSMENT

The bridge condition assessment (BCA) problem is generally regarded as a classification problem, where a rating is treated as a bridge label. However, the bridge condition ratings are ordinal. The higher the rating of bridge condition is, the more serious the bridge defects condition is. Label ranking problem extends the conventional classification and multi-label classification in the sense that it needs to predict a ranking of all class labels instead of only one or several class labels [38]. A ranking contains more label information than a single label or several unordered labels.

A. BRIDGE CONDITION ASSESSMENT MODEL

In real-world scenarios, predicting a single bridge condition rating is not enough to describe the bridge condition. Bridges with similar conditions may be assessed as different ratings because of a slight difference in attribute values. Meanwhile, noisy data can easily lead to erroneous predictions. The LR method predicts a set of ordered labels of a bridge example so that the above problem can be effectively solved.

For a sample whose bridge condition is adjacent to the rating boundary, as long as the real label appears in the first N labels in the set of the ordered labels (even if the first ranked label is wrong), it benefits the decision-making after the bridge condition rating process. The probability that the real label ranks in the top N labels can be compute as follow:

$$P_n = \sum_{i=1}^N p(M(X)_i = \lambda) \quad (5)$$

where $M(X)_i$ is the i -th ranked label that predicted by the ranker model M with the input instance X .

Fig. 2 demonstrates the architecture of the label ranking for bridge condition assessment (LR-BCA). Specifically, it applies HDC to clean the bridge condition data, and then utilizes LR as the learning algorithms. To increase robustness to residual noisy data, an ensemble model is constructed. Given a single bridge example, an ordered of labels will be predicted.

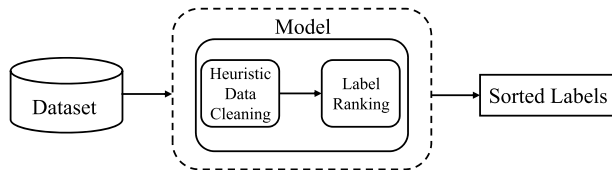


FIGURE 2. Label ranking for the bridge condition assessment.

B. LABEL RANKING

Label ranking (LR) is a key task of preference learning and its goal is to map instances into a set of ordered finite labels [38]. LR aims to learn a “label ranker” in the form of a mapping from the instance space χ to the label space ℓ . It can be seen as an extension of the traditional classification task. Unlike the traditional classification task, LR tries to map instances to an ordered set of all class labels, instead of mapping instances in instance space χ to one or more labels in label space ℓ .

In LR, $\lambda_i \succ_x \lambda_j$ denotes that instance x prefers label λ_i to λ_j , i.e., the ordering of λ_i on the instance x precedes λ_j . This preference relationship is transitive and asymmetric. Therefore, ranking can be considered as a special preference relationship. Fig. 3 shows the differences between three types of tasks, where instances space χ is $\{x_1, x_2, \dots, x_n\}$ and label space ℓ is $\{\lambda_1, \lambda_2, \dots, \lambda_c\}$.

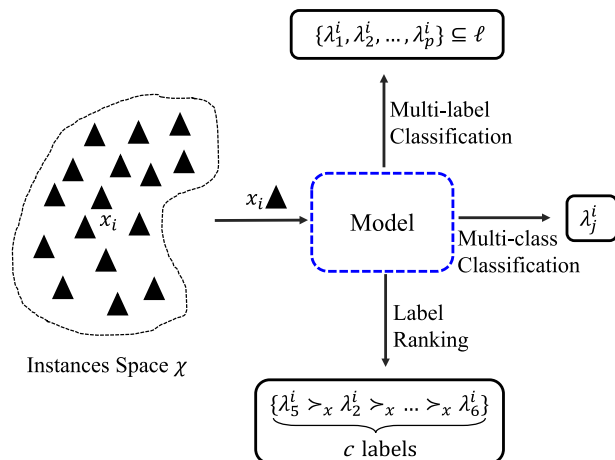


FIGURE 3. Comparison of the multi-class classification, multi-label classification and label ranking.

For example, it is possible to assume all bridges can be described in an instance space (including three attributes:

completion time, daily traffic volume, and maximum pressure). Similarly, bridge condition ratings can be described in a label space (including four labels $\lambda_1, \lambda_2, \lambda_3$, and λ_4). For a bridge instance x (2006, 76,321, 15) in instance space (years, vehicles, tons), its preference for each rating may be $\lambda_2 \succ_x \lambda_1 \succ_x \lambda_3 \succ_x \lambda_4$. Similarly, the other bridge instances have their unique preferences (i.e., a set of well-ordered labels). By training the model with different attribute values and preferences, a mapping from instances to bridge labels can be obtained.

C. SOLVE BRIDGE CONDITION ASSESSMENT AS A LABEL RANKING TASK

Given the ranking ($\lambda_2 \succ_x \lambda_1 \succ_x \lambda_3 \succ_x \lambda_4$), it is easy to know that the condition rating of the bridge instance x is located between λ_1 and λ_2 and deviates to λ_2 . For example, assume that the scores of $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are 1, 2, 3, 4, respectively. Given the real score of the bridge condition as 1.7. If the bridge condition rating is only predicted as λ_2 , the information that the bridge score is 1.7 may not be obtained. Meanwhile, if the true rating of the bridge condition is λ_1 , even if the optimal label is predicted to be λ_2 , the sub-optimal label is still the correct rating λ_1 .

There is an order relation $\lambda_3 \succ \lambda_2 \succ \lambda_1$ among three bridge ratings. Table 5 shows the transformation from a single label to a set of ordered labels. It is easy to know that bridge instance x_i with rating λ_2 prefers λ_2 comparing to λ_1 and λ_3 , but its preference between λ_1 and λ_3 cannot be determined. Due to natural order relation between bridge ratings, it is easy to know the preference of instances whose labels are λ_1 or λ_3 as shown in Table 5.

TABLE 5. Transformation between the bridge ratings and ordered labels.

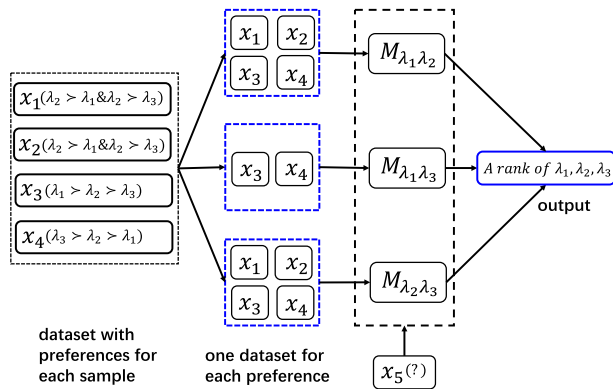
Single bridge label	A set of ordered labels
λ_1	$\lambda_1 \succ_x \lambda_2 \succ_x \lambda_3$
λ_2	$\lambda_2 \succ_x \lambda_1, \lambda_2 \succ_x \lambda_3$
λ_3	$\lambda_3 \succ_x \lambda_2 \succ_x \lambda_1$

Label ranking has been widely-studied in the literature [39]. Existing LR algorithms can be classified into three categories: reduction approaches, probabilistic approaches and tree-based approaches [40]. The reduction approaches convert the LR problem into multiple simple binary classification problems. The probabilistic approaches aim to calculate the probability of a label belonging to the instance to determine an order of all class labels. The tree-based approaches use decision tree as base algorithm. Considering the characteristic of our bridge condition assessment and the Ockham’s razor principle, we use a reduction approach in this work.

We model the BCA problem as a label ranking task, and solve it by a well-known pairwise comparison ranking algorithm. Table 6 shows the preferences of each instance, and the preference is split into three pairs of comparison labels. x_5 is the instance to be predicted. Fig. 4 shows the

TABLE 6. Examples of the bridge and its preference rating.

Instances	Ratings	Preference	$\lambda_1 > \lambda_2$	$\lambda_1 > \lambda_3$	$\lambda_2 > \lambda_3$
x_1	λ_2	$\lambda_2 > \lambda_1 \& \lambda_2 > \lambda_3$	0	/	1
x_2	λ_2	$\lambda_2 > \lambda_1 \& \lambda_2 > \lambda_3$	0	/	1
x_3	λ_1	$\lambda_1 > \lambda_2 > \lambda_3$	1	1	1
x_4	λ_3	$\lambda_3 > \lambda_2 > \lambda_1$	0	0	0
x_5	?	?	?	?	?

**FIGURE 4.** Schematic diagram of ranking by pairwise comparison.

pairwise comparison ranking algorithm on bridge condition assessment tasks. Firstly, three classifiers are trained separately using the examples with corresponding preferences. And the second step is to combine the prediction results of the three classifiers to form an ordered label set. The first label of each set of sorted labels is the rating of the bridge instance.

V. EXPERIMENTAL STUDIES

This section details the experimental studies carried out to investigate the performance of HDC and LR-BCA.

A. DATASET AND EXPERIMENTAL SETTINGS

The BCA data is provided by several highway bridge inspection agencies in China (e.g., Jiangsu Huatong Engineering Testing Co., Ltd. and Jiangsu Modern Road and Bridge Co., Ltd.). It is composed of condition data from sixteen components, such as upperparts, supports, piers, abutments, expansion joints, and drainage systems. The whole bridge condition dataset consists of 7,870 examples with 422 attributes and 3 labels.

Our proposed methods¹ are implemented by Python 3.65 32-bit running on a server with Intel(R) Core(TM) i7-6800K CPU @ 3.40GHz. We use scikit-learn 0.20 in our experiments. The algorithms and main parameters settings are shown in Table 7. While for other parameters of scikit-learn, default values are used. In the following experiments, all experiments results are obtained based on five runs of a six-fold cross-validation on the dataset. For the simplicity of expression, we record an average of the results. The widely-used metric F_1 -score is used to evaluate the performance of algorithms.

¹Both source codes and dataset will be made available in our website.

TABLE 7. Classification algorithms and corresponding parameters settings.

	Classification algorithms	Parameter setting
Stacking	KNN	n_neighbors=7, weights = 'distance'
	SVM	C=20, kernel='rbf', gamma = 'auto'
	RF	n_estimators=150, max_depth = 75
Meta classifier	RF	n_estimators=150, max_depth = 5

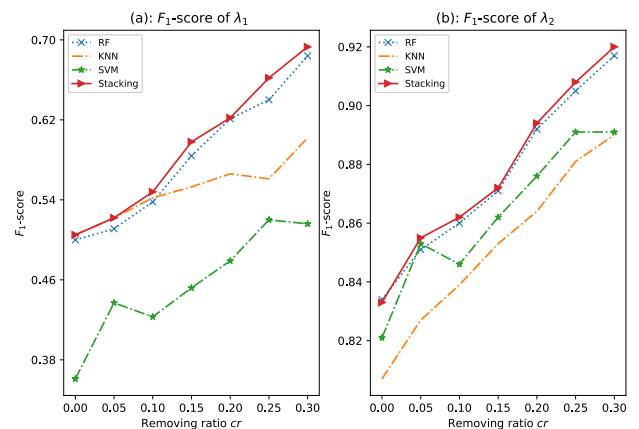
B. COMPUTATIONAL RESULTS ON HDC

To evaluate the performance of our HDC method, we have performed two groups of experiments. The goal of these experiments is to answer the following questions: (1) How does the F_1 -score vary with the cleaning ratio increases? (2) Can the HDC method still provide an advantage when it is only applied to the training set or the test set?

1) INFLUENCE OF CLEANING RATIO ON HDC

Cleaning ratio cr is an important parameter of HDC which indicates the proportion of the removed examples. This experiment aims to investigate the F_1 -score of models according to various levels of cr in the cleaning phase. We evaluate the performance of HDC by implementing four algorithms, including RF, SVM, KNN and Stacking. To demonstrate that the increase of F_1 -score is not caused by the over-fitting of maximum class, we focus on the F_1 -score of all labels. Since only few examples are available with the label λ_3 , we only consider the cases that distinguishing examples with label λ_1 and λ_2 .

Fig. 5 shows the computational results (in terms of F_1 -score) obtained from four classification algorithms. For every algorithm, HDC has an advancement of the performance on every cr level with respect to no data cleaning. Furthermore, as we can observe from Fig. 5, the increase of cr level helps the progress of F_1 -scores of both labels.

**FIGURE 5.** Comparative performance of classification algorithms with various cleaning ratios. The sub-figures (a) and (b) respectively present the experimental results by distinguishing the label λ_1 from other labels and distinguishing label λ_2 from other labels.

It indicates that HDC is not to be over-fitting on any label. Conclusion can be drawn from these results that HDC promotes the performance of models compared to no data cleaning.

2) EFFECTIVENESS OF HDC

The purpose of this group of experiments is to check that whether HDC still performs effectively when it is only applied to the training set or the testing set. We conduct two experiments (denoted as *exp1* and *exp2*) to evaluate the effectiveness of HDC. In *exp1*, we only apply HDC to the test set; In *exp2*, we only apply it to the training set. Since both the training set and testing set were noisy, if we did not clean the testing set, the incorrectly labelled examples would influence the evaluation results in test set. In a real world scenario, HDC would be applied in labelled data to generalize a clean dataset as training set, and the rating of unlabelled data would be predict by trained models.

As illustrated in Fig. 6 (a), compared with the testing accuracy without applying HDC to any dataset, the one of *exp1* has been significantly improved, up to 5.77%, 3.99%, 8.29%, and 6.15%, respectively. SVM is a noise-sensitive algorithm [6], and its predictive accuracy is greatly improved by 8.29%. The advancement of testing accuracy in *exp1* implies that in spite of the noisy training dataset, the learning algorithms can learn the spatial distribution characteristics of clean data to a certain extent because of their noise tolerance abilities. However, compared with cleaning from the entire dataset, only cleaning from the test set has a limited impact on the test accuracy.

set accurately while the noise data in the test set reduce the performance of the classifiers. Moreover, it is inevitable to eliminate a small amount of clean data in training set when performing HDC. In other words, it affects the performance of models indirectly.

In brief, the HDC method performs effectively when only cleaning from the test set and it decreases the testing accuracy when only cleaning from the training set.

C. COMPUTATIONAL RESULTS ON LR-BCA

This section presents two groups of experiments to evaluate the effectiveness of the HDC method under LR tasks, and the benefit of LR method for BCA.

1) COMPARISONS BETWEEN OUR LR METHODS UNDER VARIOUS BASELINE CLASSIFIERS

To evaluate the effectiveness of HDC in LR task, we compare the performance under three different baseline classifiers. Table 8 indicates the testing accuracy of various algorithms (Stacking, RF, KNN, SVM) in distinguishing the preference of examples between λ_i and λ_j , regarding to different *cr* levels (0%, 5%, 10%, 15%, 20%, 25%, and 30%) in the context of LR task. It is worth noting that there are a small number of examples with label λ_3 , so the result related to λ_3 may not be convincing. As illustrated in Table 8, when the *cr*

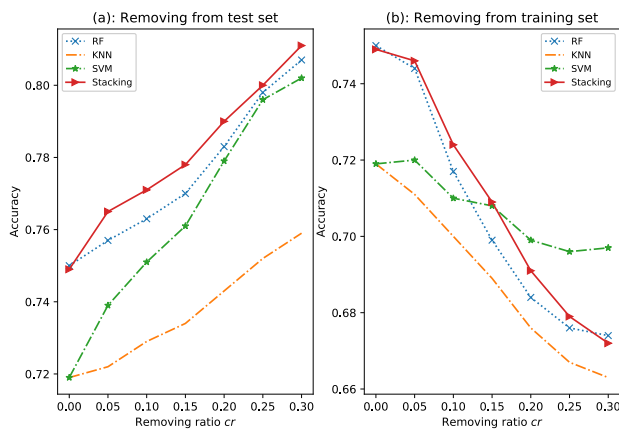


FIGURE 6. Prediction performance of various classifiers under different cleaning ratios in two cases. The sub-figures (a) and (b) presents the test accuracy of *exp1* and the test accuracy of *exp2*, respectively.

Only cleaning from the training set harms the testing accuracy of the learning algorithms as shown in Fig. 6 (b). It appears that it is counter-productive to only eliminate noise from the training set. Zhu and Wu [42] executed experiments on sixteen benchmark datasets, testing accuracy were reported to decrease on four datasets when only eliminate noise data from training set. The evidence from Fig. 6 (b) may suggest that classifiers only learn the spatial distribution of data in the training set (which is clean) while the testing set is noisy. The classifiers can identify the clean data in test

TABLE 8. Prediction performance of various classifiers under different cleaning ratios.

Cleaning ratio <i>cr</i>	Test accuracy				
	Final	RF	KNN	SVM	Stacking
0%		0.749	0.719	0.719	0.749
5%		0.771	0.765	0.744	0.775
10%		0.785	0.759	0.759	0.788
15%		0.801	0.779	0.777	0.804
20%		0.830	0.797	0.790	0.832
25%		0.847	0.820	0.811	0.854
30%		0.867	0.820	0.825	0.870
$\lambda_1 \lambda_2$	RF	KNN	SVM	Stacking	
0%	0.761	0.732	0.742	0.766	
5%	0.775	0.750	0.759	0.777	
10%	0.792	0.765	0.769	0.801	
15%	0.808	0.770	0.777	0.809	
20%	0.827	0.788	0.797	0.830	
25%	0.859	0.810	0.823	0.863	
30%	0.879	0.836	0.834	0.878	
$\lambda_1 \lambda_3$	RF	KNN	SVM	Stacking	
0%	0.950	0.952	0.943	0.960	
5%	0.956	0.946	0.937	0.961	
10%	0.962	0.953	0.941	0.966	
15%	0.966	0.956	0.952	0.969	
20%	0.960	0.948	0.946	0.968	
25%	0.966	0.952	0.949	0.968	
30%	0.965	0.968	0.958	0.975	
$\lambda_2 \lambda_3$	RF	KNN	SVM	Stacking	
0%	0.974	0.975	0.974	0.973	
5%	0.977	0.971	0.970	0.977	
10%	0.979	0.979	0.979	0.976	
15%	0.978	0.976	0.976	0.980	
20%	0.979	0.977	0.973	0.979	
25%	0.978	0.973	0.974	0.977	
30%	0.980	0.976	0.975	0.979	

level increases, the performance of classifiers increases as well. Among these base classifiers, RF achieves the best performance in all ratios of cleaning and stacking combined the advantages of three classifiers with achieving the best performance. With the application of the HDC method, all binary sub-classifiers have achieved an advancement on the performance and the final ranking accuracy has also increased.

2) BENEFIT OF LR METHOD

To demonstrate the benefit of LR, we present the Top-N test accuracy of LR with various cr level. Table 9 shows the Top-N testing accuracy of the label ranker under different levels of cr and the Top-2 testing accuracy are as high as 99%. For a bridge example that is indistinguishable between two labels, LR can indicate the rank of its preference. Considering the existence of noise data and the fuzzy boundary of two ratings, the first ranked label of the example may be wrongly predicted. However, the second ranked label is likely to be a true condition assessment label. With the implementation of LR, the error-tolerance ability of bridge condition assessment model has strengthened.

TABLE 9. Top-N test accuracy under different cleaning ratios in label ranking task.

Cleaning ratio	Top-1 accuracy	Top-2 accuracy
0%	0.760	0.987
5%	0.755	0.989
10%	0.777	0.983
15%	0.795	0.992
20%	0.818	0.989
25%	0.843	0.995
30%	0.861	0.993

VI. CONCLUSION AND FUTURE WORK

Bridge condition assessment (BCA) is an important part of bridge management systems and intelligent transportation systems. In this paper, we propose an effective label ranking approach for bridge condition assessment (LR-BCA). In addition, a heuristic data cleaning (HDC) approach is developed for filtering BCA data. Extensive experimental results on real-world dataset confirm the effectiveness of our proposed LR-BCA and HDC method. Since HDC does not use any classification algorithm as a filter, it can be combined with existing noise filters and noise-tolerant algorithms. By modeling the bridge condition assessment problem as a label ranking task, the bridge condition ratings can be predicted effectively, and the Top-2 accuracy rate is able to reach 99%. For future work, a potential research direction is to verify the effectiveness of HDC on cleaning noisy datasets in other domains.

REFERENCES

[1] S. M. Khan, S. Atamturktur, M. Chowdhury, and M. Rahman, "Integration of structural health monitoring and intelligent transportation systems for bridge condition assessment: Current status and future direction," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2107–2122, Aug. 2016.

[2] K. K. Jain and B. Bhattacharjee, "Application of fuzzy concepts to the visual assessment of deteriorating reinforced concrete structures," *J. Construct. Eng. Manage.*, vol. 138, no. 3, pp. 399–408, Mar. 2012.

[3] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann, 1993.

[4] J. Fürnkranz and G. Widmer, "Incremental reduced error pruning," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1994, pp. 70–77.

[5] J. W. W. Cohen, "Fast effective rule induction," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1995, pp. 115–123.

[6] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artif. Intell. Rev.*, vol. 33, no. 4, pp. 275–306, Apr. 2010.

[7] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[8] J. A. Sáez, M. Galar, J. Luengo, and F. Herrera, "INFFC: An iterative class noise filter based on the fusion of classifiers with noise sensitivity control," *Inf. Fusion*, vol. 27, pp. 19–32, Jan. 2016.

[9] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, May 2014.

[10] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.*, vol. 11, pp. 131–167, Aug. 1999.

[11] Y. Xiao, N. Seliya, and T. M. Khoshgoftaar, "The partitioning-and rule-based filter for noise detection," in *Proc. IEEE Int. Conf. Inf. Reuse Integr. Conf.*, Las Vegas, NV, USA, Aug. 2005, pp. 205–210.

[12] D. García-Gil, F. Luque-Sánchez, J. Luengo, S. García, and F. Herrera, "From big to smart data: Iterative ensemble filter for noise filtering in big data classification," *Int. J. Intell. Syst.*, vol. 34, no. 12, pp. 3260–3274, Dec. 2019.

[13] D. García-Gil, J. Luengo, S. García, and F. Herrera, "Enabling smart data: Noise filtering in big data classification," *Inf. Sci.*, vol. 479, pp. 135–152, Apr. 2019.

[14] B. Sluban and N. Lavrač, "Relating ensemble diversity and performance: A study in class noise detection," *Neurocomputing*, vol. 160, pp. 120–131, Jul. 2015.

[15] M. Samami, E. Akbari, M. Abdar, P. Plawiak, H. Nematzadeh, M. E. Basiri, and V. Makarenkov, "A mixed solution-based high agreement filtering method for class noise detection in binary classification," *Phys. A, Stat. Mech. Appl.*, vol. 553, Sep. 2020, Art. no. 124219.

[16] B. Sluban, D. Gamberger, and N. Lavra, "Advances in class noise detection," in *Proc. ECAI Lisbon*, Portugal, 2010, pp. 1105–1106.

[17] L. P. F. Garcia, J. A. Sáez, J. Luengo, A. C. Lorena, A. C. P. L. F. de Carvalho, and F. Herrera, "Using the one-vs-one decomposition to improve the performance of class noise filters via an aggregation strategy in multi-class classification problems," *Knowl.-Based Syst.*, vol. 90, pp. 153–164, Dec. 2015.

[18] B. Zerhari, A. A. Lahcen, and S. Mouline, "MIPCNF: Multi-iterative partitioning class noise filter," *J. Intell. Fuzzy Syst.*, vol. 37, no. 5, pp. 6761–6772, Nov. 2019.

[19] M. Sabzevari, G. Martínez-Muñoz, and A. Suárez, "A two-stage ensemble method for the detection of class-label noise," *Neurocomputing*, vol. 275, pp. 2374–2383, Jan. 2018.

[20] Z. Nematzadeh, R. Ibrahim, and A. Selamat, "Improving class noise detection and classification performance: A new two-filter CNDC model," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106428.

[21] D. Guan, K. Chen, G. Han, S. Huang, W. Yuan, M. Guizani, and L. Shu, "A novel class noise detection method for high-dimensional data in industrial informatics," *IEEE Trans. Ind. Inf.*, vol. 17, no. 3, pp. 2181–2190, Mar. 2020.

[22] M. Saito and K. C. Sinha, "Delphi study on bridge condition rating and effects of improvements," *J. Transp. Eng.*, vol. 117, no. 3, pp. 320–334, May 1991.

[23] M. Kushida and A. Miyamoto, "Modal logic to evaluate a knowledge-based bridge rating system," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 13, no. 4, pp. 227–236, Jul. 1998.

[24] Y.-M. Wang and T. M. S. Elhag, "Evidential reasoning approach for bridge condition assessment," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 689–699, Jan. 2008.

[25] S. Abu Dabous and G. Al-Khayyat, "A flexible bridge rating method based on analytical evidential reasoning and Monte Carlo simulation," *Adv. Civil Eng.*, vol. 2018, Jun. 2018, Art. no. 1290632.

- [26] M. Alsharqawi, T. Zayed, and S. Abu Dabous, "Integrated condition rating and forecasting method for bridge decks using visual inspection and ground penetrating radar," *Autom. Construction*, vol. 89, pp. 135–145, May 2018.
- [27] I. Björnsson, O. Larsson Ivanov, D. Honfi, and J. Leander, "Decision support framework for bridge condition assessments," *Structural Saf.*, vol. 81, Nov. 2019, Art. no. 101874.
- [28] T. Jiang and W. Long, "Method for evaluating technical conditions of stone arch bridge based on analytic hierarchy process," *Higw. Transp. Tech.*, vol. 32, no. 9, pp. 49–56, 2015.
- [29] H. Liu and Y. Zhang, "Bridge condition rating data modeling using deep learning algorithm," *Struct. Infrastruct. Eng.*, vol. 16, no. 10, pp. 1447–1460, Oct. 2020.
- [30] T. T. Nguyen and K. Dinh, "Prediction of bridge deck condition rating based on artificial neural networks," *J. Sci. Technol. Civil Eng.*, vol. 13, no. 3, pp. 15–25, Aug. 2019.
- [31] G. W. Flintsch and C. Chen, "Soft computing applications in infrastructure management," *J. Infrastruct. Syst.*, vol. 10, no. 4, pp. 157–166, 2004.
- [32] H. Liu, X. Wang, Y. Jiao, X. He, and B. Wang, "Condition evaluation for existing reinforced concrete bridge superstructure using fuzzy clustering improved by particle swarm optimisation," *Struct. Infrastruct. Eng.*, vol. 13, no. 7, pp. 955–965, Jul. 2017.
- [33] K. Yusuf and R. Hamid, "Developing a bridge condition rating model based on limited number of data sets," in *Bridge Engineering*. Rijeka, Croatia: IntechOpen, 2018, pp. 27–42, ch. 3, doi: [10.5772/intechopen.71556](https://doi.org/10.5772/intechopen.71556).
- [34] P. Lu, H. Wang, and D. Tolliver, "Prediction of bridge component ratings using ordinal logistic regression model," *Math. Problems Eng.*, vol. 2019, Apr. 2019, Art. no. 9797584.
- [35] Z. Li and R. Burgueño, "Structural information integration for predicting damages in bridges," *J. Ind. Inf. Integr.*, vol. 15, pp. 174–182, Sep. 2019.
- [36] P. Martínez, E. Mohamed, O. Mohsen, and Y. Mohamed, "Comparative study of data mining models for prediction of bridge future conditions," *J. Perform. Construct. Facilities*, vol. 34, no. 1, Feb. 2020, Art. no. 04019108.
- [37] L. P. F. Garcia, J. Lehmann, A. C. P. L. F. de Carvalho, and A. C. Lorena, "New label noise injection methods for the evaluation of noise filters," *Knowl.-Based Syst.*, vol. 163, pp. 693–704, Jan. 2019.
- [38] J. Fürnkranz, E. Hüllermeier, E. L. Mencia, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach. Learn.*, vol. 73, no. 2, pp. 133–153, Nov. 2008.
- [39] S. Vembu and T. Gärtner, "Label ranking algorithms: A survey," in *Preference Learning*. Berlin, German: Springer, 2011, pp. 45–64, ch. 2.
- [40] Y. Zhou and G. Qiu, "Random forest for label ranking," *Expert Syst. Appl.*, vol. 112, pp. 99–109, Dec. 2018.
- [41] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artif. Intell.*, vol. 172, nos. 16–17, pp. 1897–1916, Nov. 2008.
- [42] X. Zhu and X. Wu, "Class noise vs. Attribute noise: A quantitative study," *Artif. Intell. Rev.*, vol. 22, no. 3, pp. 177–210, Nov. 2004.



KAI WANG was born in Huangshi, Hubei, China, in 1997. He received the B.S. degree in mathematics and applied mathematics from the East China University of Science and Technology, Shanghai, China, in 2019, where he is currently pursuing the master's degree with the School of Information Science and Engineering.

His main research interests include machine learning and data mining. He is a Student Member of China Computer Federation (CCF).



TONG RUAN was born in 1973. She received the B.S. and master's degree from the East China University of Science and Technology, Shanghai, China, and the Ph.D. degree from the Institute of Software Chinese Academy of Sciences.

She is currently a Professor and a Ph.D. Supervisor with the East China University of Science and Technology. Her main research interests include knowledge graph, data mining, and data quality assessment. She is a member of China Computer Federation (CCF).



FAXIANG XIE was born in Ganjiang, Jiangsu, China, in 1977. He received the B.S. and master's degrees from the School of Transportation, Southeast University, Nanjing, China, and the Ph.D. degree in bridge engineering from Tongji University, Shanghai, China, in 2010.

He is currently an Associate Professor with the College of Civil and Transportation Engineering, Hohai University. His main research interests include Bridge engineering and concrete bridge structure.

...