IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 June 2024, accepted 16 July 2024, date of publication 23 July 2024, date of current version 31 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3432729

RESEARCH ARTICLE

Boosting Crowdsourced Annotation Accuracy: Small Loss Filtering and Augmentation-Driven Training

YANMING FU[1], WEIGENG HAN[1], JINGSANG YANG[2], HAODONG LU[1], AND XIN YU[1]
[1]School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China
[2]Liuzhou Vocational and Technical College, Liuzhou 545006, China

Corresponding author: Jingsang Yang (gxlzyjs@yeah.net)

This work was supported by the National Natural Science Foundation of China under Grant 62341210.

ABSTRACT Crowdsourcing platforms provide an efficient and cost-effective means to acquire the extensive labeled data necessary for supervised learning. However, the labels provided by untrained crowdsourcing workers often contain a considerable amount of noise. Although the application of ground truth inference algorithms to deduce integrated labels effectively enhances label quality, a certain level of noise persists. To further diminish the noise within crowdsourced labeling, this paper introduces a novel Small Loss-based Noise Correction algorithm (SLNC). SLNC first filters the crowdsourced data, leveraging the characteristic of neural networks to preferentially fits clean samples, thereby obtaining relatively clean and noisy sets. It then employs data augmentation techniques to enhance the clean set and subsequently trains the corrector on this augmented set to rectify the noisy set. SLNC has been evaluated using 16 simulated and two real-world datasets. The results indicate that SLNC surpasses comparative algorithms in the quality of the final labels.

INDEX TERMS Noise correction, crowdsourcing, data augmentation, neural network.

I. INTRODUCTION

With the widespread integration of AI technology in diverse sectors, the need for labeled data has grown significantly. However, acquiring a large volume of relevant data from domain experts often proves to be an impractical approach [1]. A feasible solution is to harness the collective power of crowdsourcing, recruiting workers to gather the needed relevant data [2]. Crowdsourcing operates as a distributed problem-solving model, where workers address large-scale tasks via an open platform [3]. The data submitted by workers, who differ in their specialization levels and personal effort, usually exhibit varying levels of accuracy [4]. Furthermore, employees driven by profit often seek to minimize their task costs, which in turn compromises the reliability of the data [5].

The research by Sheng et al. [6] provides evidence that repeated labeling can serve as an effective means to enhance label quality. Following this, ground truth inference algorithms are employed to infer the integrated label for each instance. Majority voting (MV) is one of the simplest and most effective methods for ground truth inference. However, its assumption that all crowdsourced workers are equally reliable often contradicts reality. To tackle this issue, various Weighted Majority Voting (WMV) methods have been proposed by researchers. This class of methods assigns weights to each worker or label. Tao et al. [7] addressed the inference issue as a domain adaptation problem, formulating the domain knowledge of workers with varied distributions and appropriately weighting and merging the associated noisy label sets. In a subsequent study, Tao et al. [8] introduced three weighted soft-majority voting strategies based on differential evolution. These strategies identify the optimal weights for each worker through a differential evolutionary algorithm.

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

While repeated labeling and ground truth inference algorithms [9], [10], [11] significantly enhance the quality of crowdsourced labels, the resulting integrated labels retain some level of noise [12]. In response to this problem, researchers have put forward numerous noise correction algorithms. Typically, these crowdsourced labeling noise correction algorithms [13], [14], [15] are composed of two elements: a filter and a corrector. The corrector is usually a classification model such as random trees and neural networks, among others. The filter separates the crowdsourced data into relatively clean and noisy sets, and then the corrector is trained on the clean set to rectify the noisy set. However, to the best of our knowledge, the prevailing methodologies in recent research have predominantly employed filtering based on the composition of multiple crowdsourced labels per instance, consequently neglecting the feature space inherent to the instances [12], [16], [17]. Furthermore, although the clean set obtained through filtering often comprises a limited number of instances, few studies have explored the enhancement of this set via data augmentation techniques [18], [19], [20] to improve the corrector's generalization.

To address the aforementioned issues, this paper introduces a novel Small Loss-based Noise Correction algorithm (SLNC). Initially, SLNC filters the crowdsourced data based on the size of the loss value for each instance as computed by the neural network. Subsequently, in the second stage, data augmentation techniques are employed to expand the clean set. A k-fold [21] strategy is then implemented to train multiple correctors on the enhanced clean set. When correcting the noisy set, the accuracies of these correctors on the validation set are utilized as weights to fuse the correction results.

Neural networks tend to memorize clean samples first [22], meaning that instances with smaller loss values during training are typically more reliable. However, as training progresses, the network inevitably starts to memorize noisy instances as well, leading to a gradual decrease in the loss values of these noisy instances. To address this issue, SLNC employs the cycle training method described in the literature [23].

This paper's key contributions are outlined as follows:

1) In this paper, we propose a novel crowdsourcing filtering mechanism that leverages the inherent tendency of neural networks to prioritize learning from clean samples to enable more effective filtering.
2) Given the limited number of instances in the clean set post-filtering, this paper employs data augmentation techniques to expand the set and consequently enhance the corrector's generalization ability to rectify the noisy dataset.
3) Experimental validation of the proposed algorithm was conducted using 16 simulated datasets and 2 real-world crowdsourcing datasets, demonstrating that SLNC generally outperforms the comparative algorithms.

II. RELATED WORK

As a central issue in crowdsourcing, the quality of crowdsourced labels has attracted substantial attention from a wide range of researchers. Initial efforts to enhance the quality of crowdsourced labels have centered on the application of various filtering techniques, including Classifier Filter (CF) [24], Majority Voting Filter (MVF) [25], Iterative-Partitioning Filter (IPF) [26], and Edited Nearest Neighbor (ENN) [27].

CF splits the crowdsourced dataset into n subsets, discards $1/n$ from each subset, and trains a classifier based on the remaining $n-1$ datasets. Labels that disagree with the predicted labels are considered noise and are thus excluded from the dataset. MVF processes data similarly to CF, but differs in that MVF constructs m classifiers on the remaining $n-1$ datasets. It then eliminates samples as noise when more than half the classifiers misclassify them. MVF is specifically designed to address the bias problem inherent in a single classifier. IPF partitions the crowdsourced dataset into n subsets, training a classifier on each subset independently. A sample is deemed as noise under the following conditions: 1) it is misclassified by more than half of the classifiers; 2) it is misclassified by classifiers built on the subset that includes the sample. ENN employs the KNN algorithm to identify the k nearest neighbors of each sample. If a sample's label does not match the majority of these k neighbors, it is considered noise and is removed from the dataset.

Nowadays, noise correction algorithms have become predominant, and the common ones include Polishing Labels (PL) [28], Cluster-Based Correction (CC) [28], Self-Training Correction (STC) [28], Resampling-Based Noise Correction (RNC) [29], Deep Co-teaching-based Noise Correction (DCTNC) [30], Cross-Entropy-based Noise Correction (CENC) [16], Between-class Margin-based Noise Correction (BMNC) [17], Label Distribution-based Noise Correction (LDNC) [31], Instance Difficulty-based Noise Correction (IDNC) [32], Neighbourhood-weighted Voting-based Noise Correction (NWVNC) [13], and Multi-view-based Noise Correction (MVNC) [33].

PL divides the crowdsourced data into subsets and trains a classifier on each subset. These classifiers then collectively determine the final labels for each instance through a majority voting mechanism. CC performs noise correction through iterative clustering. In each iteration, the most frequent label within a cluster is assigned to all instances in that cluster. After multiple iterations, the label that an instance receives most frequently is deemed its final label. STC begins with an initial filter to separate the dataset into clean and noisy subsets. A classifier is then trained on the clean subset and used to predict labels for the noisy subset. Instances from the noisy set that meet specific prediction criteria are transferred to the clean set, along with their predicted labels. The classifier is subsequently retrained on this updated clean set and then used to make further corrections to the remaining noisy subset. This iterative process of prediction, transfer, and retraining continues until no further instances are added to the clean set.

The RNC algorithm employs filters to split the crowdsourced data into clean and noisy subsets. Subsequently, it applies resampling techniques to these subsets repeatedly, generating several new datasets. These datasets are then used to train an ensemble of classifiers, which collectively work to rectify the labels within the noisy subset.

DCTNC recognizes that the noisy dataset may still contain some clean samples. To capitalize on this, it employs the Co-teaching [22] algorithm, which enables the learning process to focus on these potentially clean samples within the noisy set. This strategy enhances the trained classifier's ability to generalize from the noisy dataset by effectively leveraging the reliable instances it contains. CENC filters crowdsourced data by leveraging the entropy within the set of noisy labels and the predictions of classifiers. The algorithm assumes that a lower cross-entropy between the predicted class distribution and the presumed actual class distribution signifies a higher likelihood that the presumed distribution closely approximates the true class labels. BMNC is designed for noise correction in binary classification tasks with crowdsourced data, utilizing the margin in label frequency between the positive and negative classes for filtering purposes. After this initial filtering, classifiers are trained on the cleansed dataset to further correct the noisy data. In the case of LDNC, it expands upon BMNC's concept to the multiclass scenario by leveraging the margin between the frequencies of the most and second most frequent labels to perform its filtering. This adaptation allows LDNC to efficiently tackle noise correction in datasets featuring a range of classes.

In IDNC, the premise is that the reliability of labels from crowdsourced workers is linked to the complexity of the instances. To this end, IDNC proposes two strategies for evaluating the complexity of instances. Based on the outcomes of these evaluations, IDNC separates noisy from clean instances. Clean instances are used to train a classifier, which is subsequently applied to rectify the labels within the noisy set. NWVNC estimates the accuracy of an instance's aggregated label by examining the noisy labels of its neighboring instances. It then filters instances based on this estimated probability of label correctness. To correct the noisy set, three heterogeneous classifiers are developed using the clean set, which are equipped to handle the noise correction process effectively. Drawing from the principles of multi-view learning, MVNC treats the multiple noise labels associated with each instance as an additional view. It then concurrently trains classifiers on both the primary data and this secondary noise label view. This dual-view training approach enables the classifiers to more accurately correct mistakes within the noise set.

III. SMALL LOSS-BASED NOISE CORRECTION

A. PROBLEM DESCRIPTION

Labeled datasets obtained through crowdsourcing are usually denoted by $D = \{(x_i, L_i)\}_{i=1}^{N}$. Every instance x_i in the dataset corresponds to a noise label set $L_i = \{l_{ie}\}_{e=1}^{E}$. In this context, l_{ie} is the label that the crowdsourcing participant u_e provides to instance x_i, taking values from $\{c_1, c_2, \ldots, c_q\}$, where q represents the number of classes. For example, $l_{ie} = c_1$ signifies that the crowdsourcing member u_e assigns instance x_i to class c_1. $\hat{y}_i$ is an integrated label inferred from the noise set L_i corresponding to instance x_i using a ground truth inference algorithm. This paper aims to ensure the highest possible accuracy of the integrated labels $\hat{y}_i$ in the integrated dataset $\hat{D} = \{(x_i, L_i, \hat{y}_i)\}_{i=1}^{N}$.

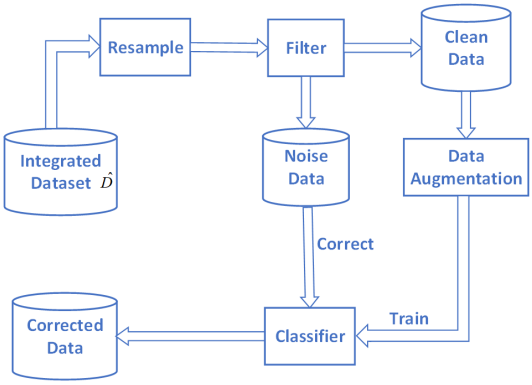

FIGURE 1. Overall framework.

B. OVERALL FRAMEWORK

The overall framework of SLNC is shown in Fig. 1. Firstly, a resampling technique is applied to the crowdsourced dataset $\hat{D}$, and multiple filters based on small loss values are constructed on the datasets obtained from multiple resamplings. Resampling [34] is employed to generate multiple diverse datasets for training the filters, which helps to improve the robustness and generalization ability of the filtering process. Resampling techniques are often used in machine learning, especially when the amount of data is small. In the context of crowdsourced datasets, resampling can help to improve the performance of filtering by creating multiple diverse datasets for training filters. In this paper, we use a put-back resampling approach, where samples are put back into the original dataset after each extraction, allowing the same data point to be selected multiple times in different resampled datasets. Let $\hat{D}_i$ denote the i-th resampled dataset, where each instance in $\hat{D}_i$ is independently drawn from $\hat{D}$ with replacement.

Then, a data augmentation technique is applied to the clean set obtained by filtering, and multiple correctors are trained on the augmented clean set to correct the noise set. Data augmentation is a technique used to increase the size and diversity of the training dataset by creating modified versions of the original data. This is particularly useful when the amount of clean data is limited, as it helps to improve the generalization ability of the trained models. The primary emphasis of this paper and its associated research does not lie in model improvement. Consequently, a straightforward three-layer DNN neural network was employed as the corrector in the experiments to minimize computational demands.

Algorithm 1 SLNC

Require: Crowdsourcing dataset $\hat{D}$
Ensure: Clean dataset D_c, Noise dataset D_n
Initialize: resample number T_r, number of filtering training epochs T_f, cyclic period difference g
for $i \leftarrow 1$ to T_r **do**
  Resample $\hat{D}$ to generate $\hat{D}_i$
  Initialize model W_i
  for $t \leftarrow 1$ to T_f **do**
    Determine the current learning rate r_t according to Eqs. (2), (3), (4)
    Train W_i on $\hat{D}_i$
    Calculate loss l_i on W_i for instances in $\hat{D}$
    Normalize l_i according to Eq. (1)
    $S_i \leftarrow S_i + l_i$
  end for
end for
According to S_i, choose top $m\%$ as $D_n i$
According to $D_n i$, employ a majority vote strategy, choose top $m\%$ as D_n, others as D_c
if a category is at risk of being entirely classified as noise **then**
  Select a subset of instances from that category with the smallest loss values
  Reclassify the selected instances into the clean set D_c
  Update the noise set D_n by removing the reclassified instances
end if
return D_c, D_n

It is important to note that for practical applications, the current model can be replaced with a more advanced alternative.

C. SLNC:FILTER

Previous studies [22] have shown that neural networks tend to memorize clean instances more easily than noisy ones, leading to higher loss values for noisy samples during training. This characteristic provides the SLNC method with a practical foundation for employing small loss values as a means to filter out noise. The specific implementation of the noise filtering process based on small loss values is described in Algorithm 1.

In the experimental section of this paper, we test the tendency of neural networks to preferentially memorize clean instances over noisy ones. We observe that the extent to which the loss values differ between noisy and clean instances varies across different datasets. In some cases, there is a marked distinction between the loss values of noisy and clean instances, while in others, the difference is less pronounced. To tackle this variability, SLNC employs a strategy where it accumulates the loss values for each instance after every training epoch. This accumulation enhances the contrast between noisy and clean instances, thereby improving the effectiveness of the filtering process.

During different stages of model training, the loss value for the same instance can vary significantly, as the model's parameters are updated and its ability to fit the data changes. Directly accumulating these loss values can obscure the differences in later stages of training, as earlier loss values are typically larger due to the model's initial poor fit to the data. To mitigate this, SLNC normalizes the loss records after each epoch of training is completed, to ensure that the difference in loss values between noisy and clean instances on each epoch is reflected in the cumulative loss values. Let X_m denote the m-th sample, Y_m denote the label of the m-th sample, $F(X_m, Y_m)$ denote the loss value of the m-th sample for the label, n be the total number of samples, and $F'(X_m, Y_m)$ be the loss value of the sample after normalization. Then the normalization operation for each epoch is represented as shown in Eq. (1).

$$F'(X_m, Y_m) = F(X_m, Y_m) - \frac{\sum_{m=1}^{n} F(X_m, Y_m)}{n} \quad (1)$$

Furthermore, SLNC takes into account that neural networks can rapidly fit simpler datasets, leading to minimal differences in loss values between noisy and clean instances after overfitting occurs. To counteract this, SLNC implements a cyclical learning rate adjustment during training. By periodically increasing the learning rate, the algorithm induces fluctuations that prevent the model from settling into an overfitted state. This strategy increases the opportunities for SLNC to observe and record the loss value differences, effectively toggling the model between overfitting and underfitting states to better identify noisy data.

Equations (2), (3), and (4) describe the cyclical variation of the learning rate. In these equations, i represents the index of the neural network, and g is a constant that accounts for the difference in cycle times between different models. The variable t denotes the t-th epoch in the training process. The maximum learning rate is given by $r_{\max}$, while $r_{\min}$ denotes the minimum learning rate. The learning rate at the t-th epoch is indicated by r_t.

$$c_i = i \cdot g \quad (2)$$

$$s_t = \frac{1 + ((t-1) \bmod c_i)}{c_i} \quad (3)$$

$$r_t = (1 - s_t) \cdot r_{\max} + s_t \cdot r_{\min} \quad (4)$$

SLNC recognizes the issue of model bias that can arise from using a single model. To mitigate this, the framework employs resampling techniques to create multiple datasets, with a separate model built on each. The $m\%$ instances with the highest loss values on each model are classified as noise sets. The final filtering decision for each instance is derived by aggregating the filtering results from all models through a voting mechanism. The top $m\%$ of instances receiving the highest number of votes are designated as the noise set, while the remaining instances constitute the clean set. In cases of a tie during voting, the instance with the highest summed

loss values across the models is designated as a noisy sample. In this study, $m\%$ serves as a hyperparameter indicating the dataset's estimated noise level. It can be practically determined by methods such as sampling. Furthermore, there is ongoing research [35] into the adaptive estimation of this noise rate.

SLNC also considers the potential presence of severe class imbalance in some datasets. To prevent the scenario where all instances of a minority category are misclassified as noisy, the framework includes a safeguard. If a category is at risk of being entirely classified as noise, SLNC selects a subset of instances from that category with the smallest loss values and reclassifies them into the clean set. This approach helps to preserve the representation of minority categories in the dataset.

D. SLNC:CORRECTOR

After obtaining the clean set D_c and the noise set D_n, data augmentation techniques are employed to enhance the minority classes within the clean set D_c.

The data augmentation technique employed by SLNC is Borderline-SMOTE [36], a specific variant of the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE [37] is designed to artificially augment the number of samples in the minority class by creating synthetic instances between existing ones. Borderline-SMOTE focuses on the minority class samples situated at the decision boundary, which are more prone to misclassification. By generating additional synthetic samples in this border region, Borderline-SMOTE helps the classifier better discern the boundary between majority and minority classes, leading to improved classification performance.

The effectiveness of Borderline-SMOTE in enhancing classifier performance has been demonstrated across various domains and datasets. In a study on software defect prediction [38], the application of Borderline-SMOTE resulted in a significant improvement in the classifier's accuracy, recall, and F1 score compared to other oversampling methods. Similarly, when applied to highly imbalanced network traffic data for intrusion detection [39], Borderline-SMOTE enabled the classifier to achieve a higher detection rate of minority class instances (i.e., network attacks) while maintaining a low false alarm rate, thereby enhancing the overall accuracy of the intrusion detection system.

After the data augmentation process using Borderline-SMOTE is completed, the clean set D_c is subjected to a k-fold cross-validation strategy to train multiple corrector models. Each corrector's performance on the validation set is evaluated to determine its accuracy, which is then used as a weighting factor in the fusion of the correction results from all the correctors. This weighted fusion methodology is detailed in Eq. (5).

$$P = \frac{\sum_{k=1}^{K} P_k \cdot Weight_k}{K} \tag{5}$$

In Eq. (5), P_k represents the probability of the category distribution predicted by model k, and $Weight_k$ represents the weight of model k, which is determined by its accuracy on the validation set. The fused category distribution, denoted by P, is computed by taking into account the individual predictions P_k and their corresponding weights $Weight_k$. If the highest probability category in P is greater than a predetermined threshold α, then a correction to the category assignment is made. Otherwise, the original integrated labeling is preserved.

IV. EXPERIMENTS

In this section, we will experimentally verify the noise correction ability of SLNC across different datasets. First, we will validate the feasibility of filtering by small loss values. Subsequently, we will assess the noise correction capability of SLNC on 16 simulated datasets and two real-world datasets, respectively. Furthermore, we will investigate the impact of varying training durations on the filtering performance of SLNC.

In our experiments, we compare SLNC with five other algorithms designed for enhancing the quality of crowdsourced labels. The following provides a brief description of each algorithm:

- **MV**: A simple yet effective approach that employs a majority voting algorithm to infer ground truth. It is commonly used as a baseline comparison algorithm in the field of crowdsourcing.
- **PL**: This approach utilizes ensemble learning to improve noise correction. It partitions the data into n subsets, trains n weak classifiers, and combines their predictions to correct the noisy labels. In our experiments, we set n to 3.
- **STC**: A self-training noise correction algorithm that retrains the corrector by placing each corrected instance into a clean set until no instance satisfies the correction condition. In our experiments, ENN was used as a filter, and the correction condition was set to a prediction score greater than 0.8.
- **DCTNC**: This approach uses the Gini impurity of instances associated with the noise set for filtering purposes. It then applies a weakly-supervised algorithm in the second phase to train the model, maximizing the utilization of the noisy data.
- **CENC**: This method filters data by comprehensively considering the entropy of the noisy label set associated with each instance and the predicted results of the classifiers. It then corrects the labels by calculating the cross-entropy between the predicted class probability distribution and the possible true class probability distribution. The number of classifiers used in our experiments is 3, and the entropy threshold is set to 0.1.
- **SLNC**: The proposed algorithm in this paper utilizes the property of neural networks to prioritize the memorization of clean instances for filtering. A data augmentation

TABLE 1. Introduction to simulation datasets.

Dataset name	Instances	Attributes	Classes
balance-scale	625	4	3
biodeg	1055	41	2
breast-w	699	9	2
car	748	4	4
credit-a	690	16	2
diabetes	768	8	2
heart-statlog	270	14	2
hypothyroid	748	28	2
iris	150	4	3
lymph	148	18	4
spambase	4601	57	2
vehicle	766	17	4
vote	435	16	2
vowel	990	13	11
wholesale	440	7	2
zoo	748	4	7

technique is employed in the corrector training phase to improve the corrector's ability to rectify noisy labels. In our experiments, the noise ratio $m\%$ is set to 20%, we use 3 correctors, and a correction threshold of $\alpha = 0.7$.

Enhancing the performance of SLNC can be achieved by moderately increasing the number of base models and resampling occurrences. However, this approach significantly raises computational demands. Therefore, following guidelines from existing literature [12], [14], [15] and practical considerations, this study sets the number of both resampling instances and base models to 3.

The SLNC filtering technique, which relies on small loss values, takes advantage of neural networks' inherent preference for fitting clean samples. This makes it particularly beneficial to use models that are designed for learning in the presence of noise, as they can effectively fit clean samples as well. However, to demonstrate the versatility of SLNC, this study conducts experiments using a simple MLP model.

The parameter $m\%$, representing the estimated noise rate, is determined through a noise rate estimation algorithm. Recognizing that accurately estimating the noise rate can be challenging, and in line with parameter settings found in related work [30], we have set $m\%$ to 20%, a value commonly employed across similar studies.

A. SMALL LOSS FILTRATION FEASIBILITY VERIFICATION

This section aims to investigate the feasibility of filtering noisy instances based on small loss values. We obtained 16 datasets from the CEKA platform [40], which are widely used in simulation experiments for crowdsourcing noise correction. Their widespread use is attributed to the datasets' diverse numbers of categories, features, and samples, providing a strong representative quality. Table 1 provides a brief description of each dataset.

In this experiment, we randomly modified the labels of 40% of the instances to simulate mislabeling. We then tracked the changes in the average loss values of both the label-modified instances and the unmodified instances throughout the training process. Fig. 2 presents the results of this experiment.

The experimental results clearly demonstrate that the average loss value of the noisy instances is significantly higher than that of the clean instances. This finding confirms the feasibility of noise filtering by monitoring the loss value level of each instance during the training process.

B. NOISE CORRECTION EXPERIMENTS ON SIMULATED DATASETS

This section focuses on simulating the crowdsourcing annotation process using the 16 datasets presented in Table 1. During the experiment, the ground truth labels of the instances will be concealed. Each instance will be assigned 8 crowdsourcing labels, which include both correct and incorrect annotations. To better simulate the crowdsourcing scenario, we introduce two methods for generating crowdsourcing noise labels: Gaussian noise and bias noise. The variance of all Gaussian distributions used in the experiment is set to 0.1, with a mean of 0.6. Within the scope of this research, label quality is quantitatively assessed by the proportion of the corrected labels that align with the corresponding ground truth labels of the instances. The two noise label generation methods are described below:

Gaussian Noise: This is the most prevalent method for generating noisy label sets in crowdsourced simulation datasets. Each worker has a probability P of producing the correct label and a $1-P$ probability of generating an incorrect label. The probability P follows the Gaussian distribution, as detailed in Eq. (6).

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \tag{6}$$

Bias Noise: This method simulates the influence of worker bias in crowdsourcing settings. For instance, given the same image, some participants in a crowdsourcing task might label it as 'interesting', while others may deem it 'boring'. Each worker has a probability P of generating the correct labels and a $1-P$ probability of generating incorrect labels, where P adheres to the Gaussian distribution. Incorrect labels are generated according to Eq. (7), where T represents the ground truth label, C denotes the number of label categories, and N signifies the generated noisy labels.

$$N = (T+1) \bmod C \tag{7}$$

Table 2 offers an in-depth comparison of label quality under Gaussian Noise. The arithmetic mean of each method's performance across all datasets is presented in the last row, summarizing their overall effectiveness. Utilizing the data from Table 2, the Wilcoxon [41] signed-rank test is applied to conduct a pairwise statistical comparison of the methods. The results of these comparisons are detailed in Table 3, which adheres to the widely accepted significance level of $\alpha = 0.05$.

Table 4, in a similar fashion, presents the comparative analysis of label quality affected by Bias Noise. Here too,

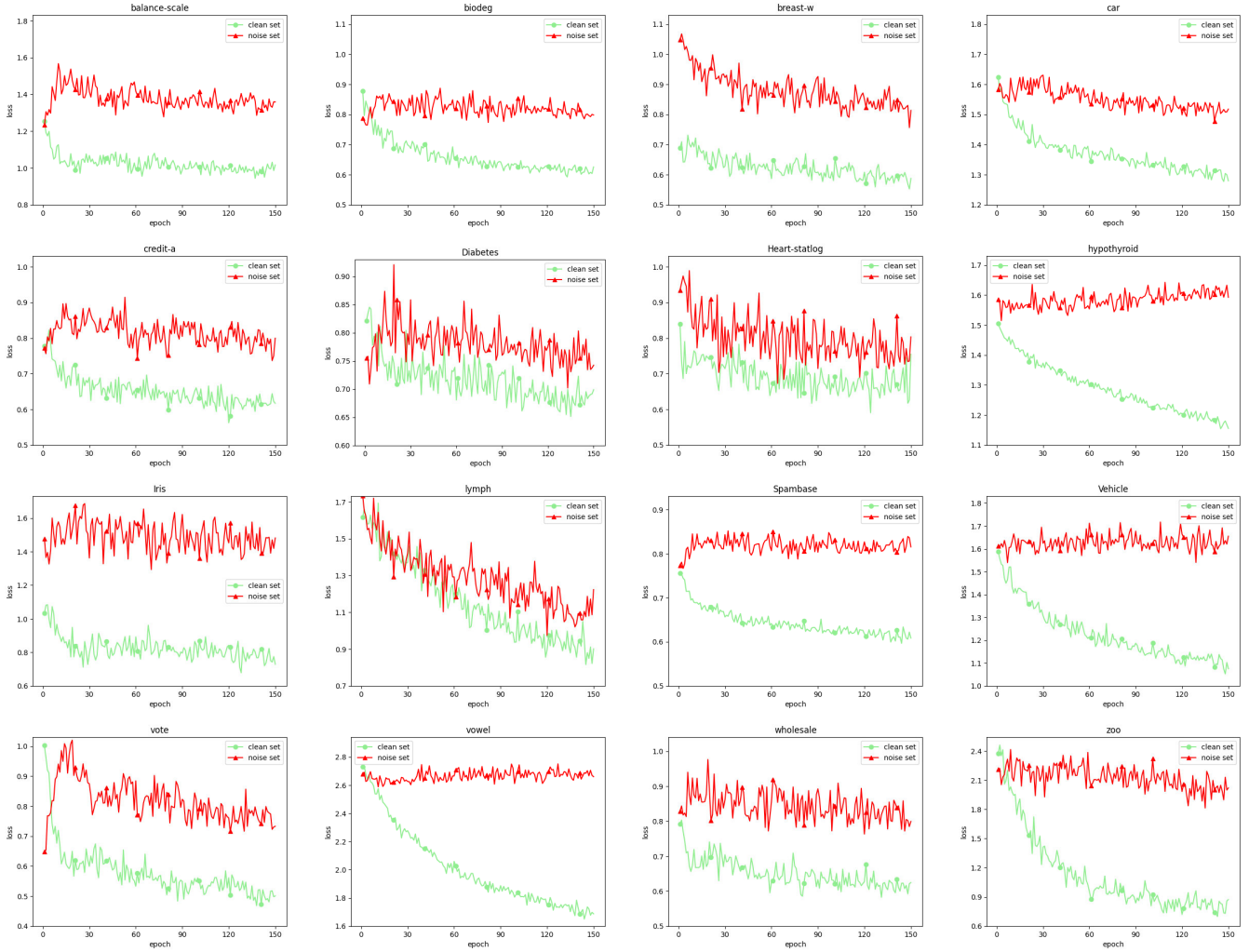

FIGURE 2. Loss change record for different datasets during training.

the last row provides the arithmetic mean for each method's label quality across the datasets, serving as an indicator of their overall performance. Comparative analysis based on the results in Table 4 is performed through the Wilcoxon signed-rank test. Table 5 presents these results, maintaining the widely accepted significance level of $\alpha = 0.05$.

Within Tables 2 and 4, the highest label quality score for each dataset is emphasized in **bold**. In Tables 3 and 5, the symbol • indicates a statistically significant better performance of the method in the row relative to the method in the corresponding column.

The experimental results presented in this study comprehensively validate the effectiveness of SLNC in enhancing the quality of crowdsourced labels. Based on these findings, we can draw the following key conclusions:

1) For both Gaussian and Bias noise, the label quality corrected and processed by SLNC is significantly higher than that of the other five widely used comparison algorithms.

TABLE 2. Gaussian noise experiment results.

Dataset	MV	PL	STC	DCTNC	CENC	SLNC
balance-scale	79.84	81.92	90.56	84.32	91.68	**92.64**
biodeg	73.27	64.83	73.36	73.84	73.84	**78.39**
breast-w	70.67	86.70	75.68	73.96	85.12	**87.70**
car	92.07	79.40	77.20	**94.10**	89.18	93.46
credit-a	67.83	46.23	50.14	67.54	52.32	**68.12**
diabetes	72.92	69.92	67.19	73.70	71.88	**76.56**
heart-statlog	72.59	64.81	62.59	75.93	69.63	**77.04**
hypothyroid	93.35	94.64	95.12	**96.29**	95.73	95.52
Iris	86.67	96.67	96.67	91.33	98.00	**98.67**
lymph	89.86	77.03	77.03	**91.32**	79.73	91.22
spambase	73.16	59.60	67.27	74.12	67.16	**75.72**
vehicle	91.49	73.17	66.31	**93.75**	77.30	91.49
vote	72.64	83.68	72.87	77.47	88.74	**88.97**
vowel	96.57	65.96	79.90	**96.97**	82.02	96.57
wholesale	74.55	84.09	76.59	77.50	86.14	**87.50**
zoo	92.08	86.14	79.21	94.24	95.05	**96.04**
average	81.22	75.92	75.48	83.52	81.47	**87.23**

2) According to the Wilcoxon test at the widely accepted significance level of $\alpha = 0.05$, SLNC significantly

TABLE 3. Wilcoxon signed-rank test results for gaussian noise.

	MV	PL	STC	DCTNC	CENC	SLNC
MV	-					
PL		-				
STC			-			
DCTNC	•	•	•	-		
CENC		•	•		-	
SLNC	•	•	•	•	•	-

TABLE 4. Bias noise experiment results.

Dataset	MV	PL	STC	DCTNC	CENC	SLNC
balance-scale	71.20	63.52	65.92	71.04	73.76	**76.80**
biodeg	73.55	63.60	72.04	74.12	72.51	**81.23**
breast-w	73.96	90.84	81.40	77.54	91.27	**92.27**
car	81.25	72.45	78.41	80.38	86.28	**88.37**
credit-a	68.12	50.29	52.61	**70.14**	64.20	67.97
diabetes	74.35	70.18	66.28	74.74	74.74	**76.30**
heart-statlog	73.33	66.67	62.59	74.07	65.56	**75.56**
hypothyroid	81.65	95.36	94.72	81.12	95.84	**97.06**
Iris	76.00	62.00	81.33	78.67	**94.00**	90.00
lymph	83.11	77.70	75.00	82.38	77.70	**85.81**
spambase	73.20	60.27	67.01	74.55	70.42	**75.87**
vehicle	**75.65**	66.43	57.45	71.77	66.19	**75.65**
vote	74.94	88.97	87.13	79.54	**92.18**	91.03
vowel	**79.70**	52.63	46.06	79.29	74.14	**79.70**
wholesale	75.45	86.82	82.73	78.18	86.36	**87.50**
zoo	80.20	80.20	80.20	80.14	**88.12**	87.13
average	75.98	71.75	71.93	76.73	79.58	**83.02**

TABLE 5. Wilcoxon signed-rank test results for bias noise.

	MV	PL	STC	DCTNC	CENC	SLNC
MV	-					
PL		-				
STC			-			
DCTNC				-		
CENC		•	•		-	
SLNC	•	•	•	•	•	-

outperforms five other well-established crowdsourcing label quality improvement algorithms in terms of label quality.

3) Surprisingly, the average label quality reveals that the performance of PL and STC is lower than that of the baseline algorithm MV. This can be attributed to the fact that both algorithms heavily rely on the classifier, and the classifier used in the experiments in this paper is a simple three-layer DNN model, which may not exhibit outstanding performance.

C. NOISE CORRECTION EXPERIMENTS ON REAL-WORLD DATASETS

In this section, we will evaluate the label correction performance of SLNC on two real-world datasets, Music [42] and LabelMe [43]. The ground truth labels of both datasets are known, and the ground truth labels are hidden before publishing the datasets to real crowdsourcing platforms for labeling.

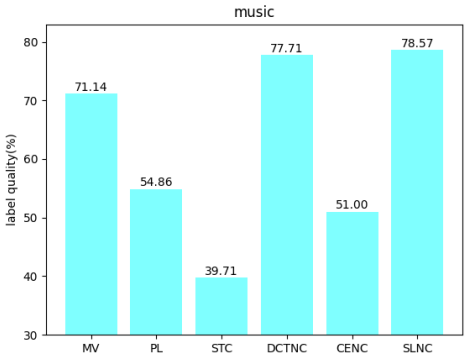

FIGURE 3. Music label quality.

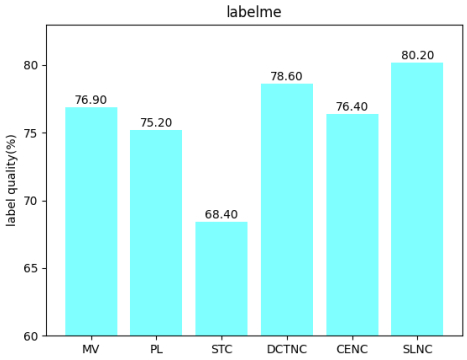

FIGURE 4. LabelMe label quality.

Music is a music categorization dataset with a total of 700 samples, encompassing 10 categories such as "blues", "classical", "country", "disco", "hiphop", "jazz", "metal", "pop", "reggae", and "rock". Each sample is characterized by 123 feature attributes. A total of 2,945 labels are obtained from the Amazon Mechanical Turk (AMT) crowdsourcing platform, with an average of 4.21 labels per instance.

LabelMe is an image classification dataset sourced from the LabelMe crowdsourcing platform. It consists of 1,000 images spanning a diverse set of categories, including "highway", "inner city", "high rise building", "street", "forest", "coast", "mountain", and "wilderness". On average, each image in the dataset has been assigned 2.49 labels by the crowdsourcing community.

Figures 3 and 4 present a detailed comparison of the label quality among the six noise correction methods: MV, PL, STC, DCTNC, CENC, and SLNC. The results clearly demonstrate that SLNC outperforms the other methods. On the Music dataset, SLNC achieves a label quality of 78.57%, surpassing MV (71.14%), PL (54.86%), STC (39.71%), DCTNC (77.71%), and CENC (51%). Likewise, on the LabelMe dataset, SLNC exhibits a higher label quality (80.20%) compared to MV (76.90%), PL (75.20%), STC (68.40%), DCTNC (78.60%), and CENC (76.40%). These comparative results lead to conclusions that are consistent with those drawn from the simulated dataset.

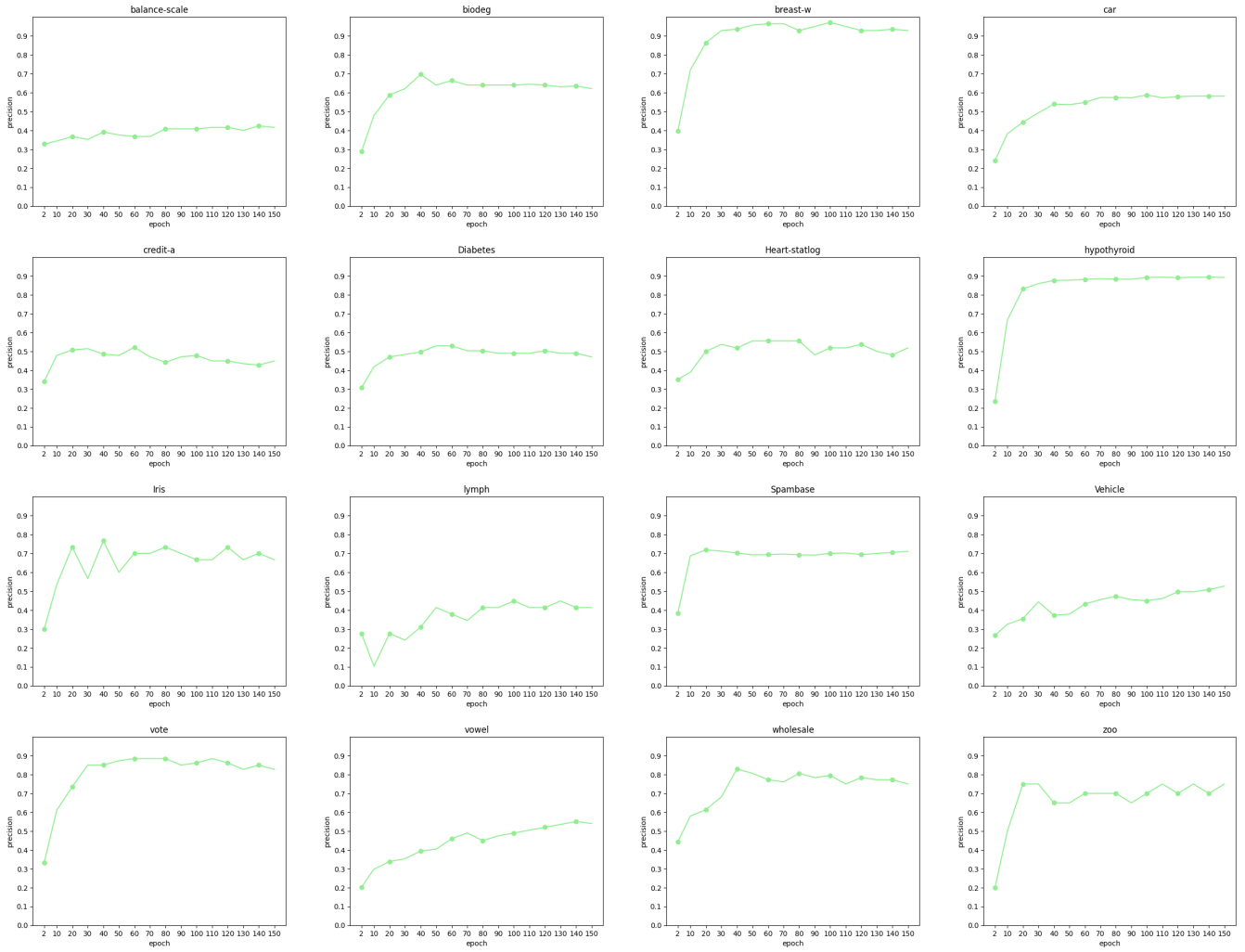

FIGURE 5. Filtering accuracy change record for different datasets during training.

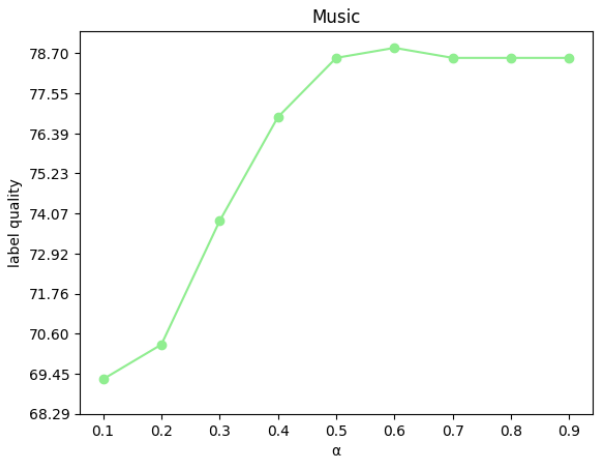

FIGURE 6. The effect of α on label quality in the Music dataset.

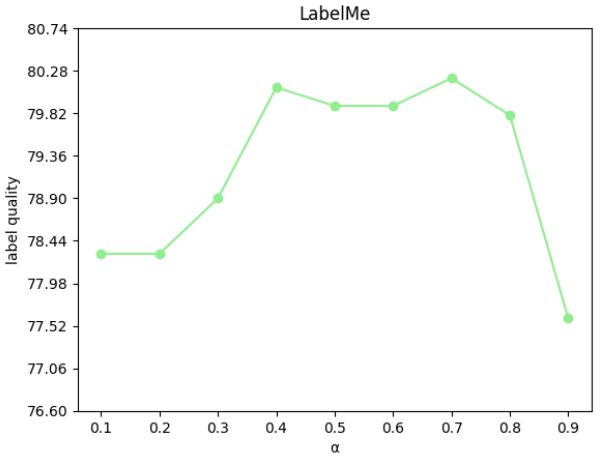

FIGURE 7. The effect of α on label quality in the LabelMe dataset.

D. PARAMETER SETTING EXPERIMENTS

In this section, we tested the effect of the number of training epochs on the filtering performance in 16 simulated datasets. We also tested the effect of the correction threshold α on the final label quality in two real-world datasets. We define filtering accuracy as the ratio of correctly identified noisy instances (i.e., instances that are correctly classified as noisy

by the filter) to the total number of instances classified as noisy by the filter. Label quality is defined as the proportion of corrected labels that match the ground truth label of the instance.

The experiments on the effect of α on the final label quality, as shown in Fig.6 and Fig.7, demonstrate that different α values have a significant impact on the label quality. Specifically, as α increases, the label quality initially improves, reaches a peak, and then declines, exhibiting a hump-shaped pattern. This suggests that an optimal value of α exists for achieving the highest label quality. Based on these observations and to obtain favorable results across a wider range of datasets, we set α to 0.7 in our experiments.

However, as demonstrated by the experimental results presented in Figure 5, we observe that SLNC maintains a more stable filtering performance and does not suffer from a significant decrease in effectiveness due to overfitting caused by excessive training times. To ensure optimal filtering performance across different datasets, we set the number of training epochs to a relatively large value of 100 in this study.

V. CONCLUSION AND FUTURE WORK

In this study, we introduce a novel crowdsourcing noise correction algorithm called Small Loss Noise Correction (SLNC). SLNC leverages the inherent property of neural networks to preferentially learn from clean samples, enabling effective filtering of noisy instances. Furthermore, it incorporates ensemble learning techniques to enhance the performance of individual filters. During the corrector training phase, SLNC employs data augmentation methods to expand the clean set obtained from the filtering process and utilizes the k-fold cross-validation strategy to train multiple correctors within the clean set for correcting the noise set. The effectiveness of the proposed algorithm is experimentally validated on 16 simulated datasets and 2 real-world datasets.It is important to note that SLNC relies on the learning ability of the neural network, but when the single label noise rate is greater than fifty percent the neural network is not able to learn the correct knowledge, and at this point SLNC does not work correctly.

For future research, we intend to develop a simple and easy-to-implement adaptive noise rate estimation algorithm to improve the practicality of SLNC in real-world applications. Additionally, as the correction results of SLNC are sensitive to the parameter α, and an appropriate α setting can effectively improve the label quality, we plan to propose an adaptive α parameter setting method in future work.

REFERENCES

[1] S. Li, X. Bai, and S. Wei, "Blockchain-based crowdsourcing framework with distributed task assignment and solution verification," *Secur. Commun. Netw.*, vol. 2022, pp. 1–16, Mar. 2022.
[2] J. Zhang, X. Wu, and V. S. Sheng, "Learning from crowdsourced labeled data: A survey," *Artif. Intell. Rev.*, vol. 46, no. 4, pp. 543–576, Dec. 2016.
[3] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar, "Quality control in crowdsourcing systems: Issues and directions," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 76–81, Mar. 2013.
[4] M. Wu, Q. Li, F. Yang, J. Zhang, V. S. Sheng, and J. Hou, "Learning from biased crowdsourced labeling with deep clustering," *Expert Syst. Appl.*, vol. 211, Jan. 2023, Art. no. 118608.
[5] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Netw.*, vol. 32, no. 4, pp. 54–60, Jul. 2018.
[6] V. S. Sheng, F. Provost, and P. G. Ipeirotis, "Get another label? Improving data quality and data mining using multiple, noisy labelers," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 614–622.
[7] D. Tao, J. Cheng, Z. Yu, K. Yue, and L. Wang, "Domain-weighted majority voting for crowdsourcing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 163–174, Jan. 2019.
[8] F. Tao, L. Jiang, and C. Li, "Differential evolution-based weighted soft majority voting for crowdsourcing," *Eng. Appl. Artif. Intell.*, vol. 106, Nov. 2021, Art. no. 104474.
[9] X. Gao, H. Huang, C. Liu, F. Wu, and G. Chen, "Quality inference based task assignment in mobile crowdsensing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3410–3423, Oct. 2021.
[10] Y. Liu, F. Liu, H.-T. Wu, J. Yang, K. Zheng, L. Xu, X. Yan, and J. Hu, "RPTD: Reliability-enhanced privacy-preserving truth discovery for mobile crowdsensing," *J. Netw. Comput. Appl.*, vol. 207, Nov. 2022, Art. no. 103484.
[11] W. Mo, Z. Li, Z. Zeng, N. N. Xiong, S. Zhang, and A. Liu, "SCTD: A spatiotemporal correlation truth discovery scheme for security management of data platform," *Future Gener. Comput. Syst.*, vol. 139, pp. 109–125, Feb. 2023.
[12] Y. Dong, L. Jiang, and C. Li, "Improving data and model quality in crowdsourcing using co-training-based noise correction," *Inf. Sci.*, vol. 583, pp. 174–188, Jan. 2022.
[13] H. Li, L. Jiang, and S. Xue, "Neighborhood weighted voting-based noise correction for crowdsourcing," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 7, pp. 1–18, Aug. 2023.
[14] L. Ren, L. Jiang, and C. Li, "Label confidence-based noise correction for crowdsourcing," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105624.
[15] X. Wu, L. Jiang, W. Zhang, and C. Li, "Three-way decision-based noise correction for crowdsourcing," *Int. J. Approx. Reasoning*, vol. 160, Sep. 2023, Art. no. 108973.
[16] W. Xu, L. Jiang, and C. Li, "Improving data and model quality in crowdsourcing using cross-entropy-based noise correction," *Inf. Sci.*, vol. 546, pp. 803–814, Feb. 2021.
[17] C. Li, L. Jiang, and W. Xu, "Noise correction to improve data and model quality for crowdsourcing," *Eng. Appl. Artif. Intell.*, vol. 82, pp. 184–191, Jun. 2019.
[18] M. Bayer, M.-A. Kaufhold, and C. Reuter, "A survey on data augmentation for text classification," *ACM Comput. Surveys*, vol. 55, no. 7, pp. 1–39, Jul. 2023.
[19] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.
[20] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *J. Big Data*, vol. 8, no. 1, p. 101, Dec. 2021.
[21] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Stat. Soc. B, Stat. Methodology*, vol. 36, no. 2, pp. 111–133, Jan. 1974.
[22] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 8527–8537.
[23] J. Huang, L. Qu, R. Jia, and B. Zhao, "O2U-Net: A simple noisy label detection approach for deep neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3325–3333.
[24] D. Gamberger, N. Lavrac, and C. Groselj, "Experiments with noise filtering in a medical domain," in *Proc. ICML*, vol. 99, 1999, pp. 143–151.
[25] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.*, vol. 11, pp. 131–167, Aug. 1999.
[26] T. M. Khoshgoftaar and P. Rebours, "Improving software quality prediction by noise filtering techniques," *J. Comput. Sci. Technol.*, vol. 22, no. 3, pp. 387–396, May 2007.
[27] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 3, pp. 408–421, Jul. 1972.

[28] B. Nicholson, V. S. Sheng, and J. Zhang, "Label noise correction and application in crowdsourcing," *Expert Syst. Appl.*, vol. 66, pp. 149–162, Dec. 2016.
[29] W. Xu, L. Jiang, and C. Li, "Resampling-based noise correction for crowdsourcing," *J. Exp. Theor. Artif. Intell.*, vol. 33, no. 6, pp. 985–999, Nov. 2021.
[30] K. Zhu, S. Xue, and L. Jiang, "Improving label quality in crowdsourcing using deep co-teaching-based noise correction," *Int. J. Mach. Learn. Cybern.*, vol. 14, no. 10, pp. 3641–3654, Oct. 2023.
[31] Z. Chen, L. Jiang, and C. Li, "Label distribution-based noise correction for multiclass crowdsourcing," *Int. J. Intell. Syst.*, vol. 37, no. 9, pp. 5752–5767, Sep. 2022.
[32] Y. Hu, L. Jiang, and C. Li, "Instance difficulty-based noise correction for crowdsourcing," *Expert Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118794.
[33] X. Li, C. Li, and L. Jiang, "A multi-view-based noise correction algorithm for crowdsourcing learning," *Inf. Fusion*, vol. 91, pp. 529–541, Mar. 2023.
[34] C. Tantithamthavorn, A. E. Hassan, and K. Matsumoto, "The impact of class rebalancing techniques on the performance and interpretation of defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 46, no. 11, pp. 1200–1219, Nov. 2020.
[35] M. Zhu, L. Zhang, L. Wang, D. Li, J. Zhang, and Z. Yi, "Robust co-teaching learning with consistency-based noisy label correction for medical image classification," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 18, no. 4, pp. 675–683, Nov. 2022.
[36] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.* Berlin, Germany: Springer, 2005, pp. 878–887.
[37] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[38] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: Improving classification performance when training data is skewed," in *Proc. 19th Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.
[39] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: An overview," *Prog. Artif. Intell.*, vol. 1, no. 1, pp. 89–101, Apr. 2012.
[40] J. Zhang, V. S. Sheng, B. Nicholson, and X. Wu, "CEKA: A tool for mining the wisdom of crowds," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 2853–2858, 2015.
[41] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.
[42] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, May 2008.
[43] F. Rodrigues, F. Pereira, and B. Ribeiro, "Learning from multiple annotators: Distinguishing good from random labelers," *Pattern Recognit. Lett.*, vol. 34, no. 12, pp. 1428–1436, Sep. 2013.

YANMING FU received the Ph.D. degree from Sichuan University, in 2011. He is currently an Associate Professor with Guangxi University. He has published a number of journals and conference papers. His research interests include data mining, computation intelligence, and network security.

WEIGENG HAN is currently pursuing the degree with Guangxi University. His research interests include mobile crowd-sensing and weakly supervised learning.

JINGSANG YANG is currently a Faculty Member specializing in data mining research with Liuzhou Vocational and Technical College.

HAODONG LU is currently pursuing the degree with Guangxi University. His research interests include mobile crowd-sensing and privacy-preserving computation.

XIN YU received the B.Sc. degree from the Central South University of Technology, Changsha, China, in 1995, and the M.Sc. and Ph.D. degrees from Central South University, Changsha, in 2001 and 2007, respectively. He is currently a Professor with Guangxi University, Nanning, China. His current research interests include artificial neural network theory and optimization.