Method 1: Confidence Score Calculation using Weighted Sum

In a standard voting system, for every instance i and class k, each contributor in the group—whether a worker in a crowd or a model in an ensemble learning context—provides a class label. The label receiving the most votes, meaning it is predicted by the majority of contributors, is selected as the final prediction. This approach is often referred to as majority voting or hard voting. This method could be improved by taking into account not only the number of votes each label receives, but also the confidence associated with each vote. This introduces the notion of a "weighted sum of all votes for a particular class". As part of this study, we propose techniques that assign a weight to each contributor in the group. This calculation is based on their voting consistency and the degree to which they concur with their peers. To compute the weighted sum of all votes for each class, we can combine the calculated weights with the corresponding labels, whether provided or predicted. This calculation gives greater confidence to votes with greater certainty. This refined approach prioritizes certainty, thereby enhancing the ensemble's overall effectiveness. The confidence score $F_{\Omega}^{(i,k)}$ is formulated as follows.

$$F^{(i,k)} = F_{\Omega}^{(i,k)} = \sum\nolimits_{\alpha=1}^{M} \omega_{\alpha}^{(k)} \delta\left(\eta_{\alpha}^{(i,k)}\ ,\ \ \nu^{(i,k)}\right) \tag{1}$$

where δ is the Kronecker delta function.

Method 2: Confidence Score Calculation using CDF of Beta Distribution Function

The binomial distribution survival function, also referred to as the complementary cumulative distribution function (CCDF), provides the probability of observing a result as extreme or more extreme than a given value. It provides the probability that a random variable drawn from the binomial distribution is greater than or equal to a given value. It can be applied in the following ways when calculating confidence scores:

1. **Hypothesis testing:** Suppose we want to test a hypothesis concerning a parameter of a population, using a set of observed samples. Given the null hypothesis, the binomial survival function can be used to calculate the probability of observing a result as extreme or more extreme than the one observed. This probability is the p-value, and if it is very small, the null hypothesis may be rejected. In this context, the confidence score could be interpreted as (1 − p-value), a measure of the certainty with which the null hypothesis can be rejected.
2. **Binary classification:** Consider a binary classification problem in which an algorithm sorts objects into two groups. Given the observed results, the binomial survival function can be used to calculate the probability of misclassification. The confidence score in this instance could be interpreted as (1 − probability of misclassification).

The confidence score serves as a quantitative measure of the reliability of the classification of instance 'i' into class 'k'. The CDF of the beta distribution function can be defined as follows [**?**]:

$$f(x;\ l,u) = \frac{B(x;\ l,u)}{B(l,u)} = I_x(l,u) \tag{2}$$

where $B(x;\ l,u)$ is the incomplete beta function, and $I_x(l,u)$ is the regularized incomplete beta function. l and u terms are typically shown as α and β, but we use different notations in this paper to avoid confusion with the annotator index α and confidence score measurement beta.

The CDF of the beta distribution at the decision threshold of $x = 0.5$ (denoted as $I_{0.5}$) is used to calculate the confidence score $F_{\beta}^{(i,k)}$. To calculate $I_{0.5}$, we first need to calculate two shape parameters $l^{(i,k)}$ and $u^{(i,k)}$

Shape Parameters:

Two shape parameters $l^{(i,k)}$ and $u^{(i,k)}$ of the Beta distribution are calculated as follows:

$$\begin{aligned} l^{(i,k)} &= 1 + \sum_{\alpha=1}^{M} \omega_\alpha^{(k)}\ \delta\left(\eta_\alpha^{(i,k)}, \nu^{(i,k)}\right) \\ u^{(i,k)} &= 1 + \sum_{\alpha=1}^{M} \omega_\alpha^{(k)}\ \delta\left(\eta_\alpha^{(i,k)}, 1-\nu^{(i,k)}\right) \end{aligned} \tag{3}$$

A Major difference between our calculations shown in Eq. 3 and Tao [7], is that, Tao calculates a weighted sum of the annotators' labels ($z_\alpha^{(i,k)}$) that have voted on the positive class ($\delta(z_\alpha^{(i,k)}, +)$) or negative class ($\delta(z_\alpha^{(i,k)}, -)$). However, we take the predicted labels obtained from the trained classifiers belonging to each annotator ($\eta_\alpha^{(i,k)}$) instead of annotators' labels ($z_\alpha^{(I,k)}$). Also instead of calculating the weighted sum of all positive and negative labels, we calculate the weighted sum of all annotators' predicted labels that are the same as the calculated aggregated label (Eq. **??**), ($\delta(\eta_\alpha^{(i,k)}, \nu^{(i,k)})$) and differs from it ($\delta(\eta_\alpha^{(i,k)}, 1-\nu^{(i,k)})$) respectively. Here, the shape parameters are effectively a weighted sum (with weights $\omega_\alpha^{(k)}$) of all the correct and incorrect aggregated labels, modulated by a Dirac delta function δ, which acts to selectively include terms where the condition inside the delta function is satisfied.

Confidence Score:

The confidence score is subsequently calculated utilizing the previously determined shape parameters, as follows:

$$F^{(i,k)} = F_\beta^{(i,k)} = I_{0.5}\left(l^{(i,k)}, u^{(i,k)}\right) \tag{4}$$

$$= \sum_{t=[l^{(i,k)}]}^{T-1} \frac{(T-1)!}{t!(T-1-t)!} 0.5^{T-1} \tag{5}$$

where $T = \left[l^{(i,k)} + u^{(i,k)}\right]$ and $[\cdot]$ denotes rounding to the nearest integer.

0.1 Metrics

- **Accuracy:** The accuracy of the model is the proportion of true results (both true positive and true negatives) among the total number of cases examined. Mathematically, accuracy can be represented as

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\delta\left(\nu^{(i,k)}, y^{(i,k)}\right) \tag{6}$$

  where δ is the Kronecker delta function, N is the total number of instances, and K is the number of classes, and $y^{(i,k)}$ and $\nu^{(i,k)}$ are the ground truth and aggregated label, respectively, for class k and instance i. Accuracy is most effective for balanced classes, and its interpretation can be skewed in the presence of significant class imbalance.

- **F1 Score:** The F1 score is the harmonic mean of precision and recall and can be used for assessing the quality of aggregated labels, especially in the presence of imbalanced classes. F1 score provides a balanced measure of precision and recall, ranging from 0 to 1, where 1 represents the best possible F1 score. It is computed as

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

  where $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$ and $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$, and TP, FP and FN are the numbers of true positives, false positives and false negatives, respectively.

- **Area Under the Curve for the Receiver Operating Characteristic (AUC-ROC):** AUC-ROC measures the trade-off between the true positive rate (sensitivity) and false positive rate (1-specificity). Higher AUC-ROC values indicate better classification performance.

- **Brier Score:** Brier score provides a measure of the accuracy of the probabilistic (or confidence score) predictions. It is calculated as the mean squared error between the estimated confidence score $F^{(i,k)}$ and the ground truth label $y^{(i,k)}$, thereby rewarding more confident predictions. It can be calculated as follows:

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \left(F^{(i,k)} - y^{(i,k)} \right)^2 \tag{8}$$

- **Expected Calibration Error (ECE):** The ECE quantifies the calibration of confidence scores produced by a model. It is computed as a weighted average of the absolute differences between the actual accuracies and the predicted confidences within each bin when predictions are grouped into distinct bins based on their predicted confidence. A lower ECE signifies a model whose predicted probabilities closely match the observed frequencies across all bins. ECE can be formulated as follows:

$$\text{ECE} = \sum_{b=1}^{B} \frac{|B_b|}{N} \left| \text{Accuracy}(B_b) - \text{Confidence-Score}(B_b) \right| \tag{9}$$

where B is the number of bins, B_b is the set of instances in bin b, N is the total number of instances, Accuracy(B_b) is the accuracy of bin b, and Confidence-Score(B_b) is the average confidence of bin b.

1 Results

To evaluate our proposed technique, we conducted a series of experiments comparing the proposed technique with several existing techniques such as MV, Tao [7], and Sheng [6], as well as with other crowdsourcing methodologies reported in the crowd-kit package [9] including Gold Majority Voting, MMSR [5], Wawa, Zero-Based Skill, GLAD [10], and Dawid Skene [2].

1.1 Datasets

We report the performance of our proposed techniques on various datasets. These datasets cover a wide range of domains and have varying characteristics in terms of the number of features, samples, and class distributions. Table 1 provides an overview of the datasets used. All datasets are obtained from the University of California, Irvine (UCI) repository [3].

Table 1 Description of the datasets used.

Dataset	#Features	#Samples	#Positives	#Negatives
kr-vs-kp	36	3196	1669	1527
mushroom	22	8124	4208	3916
iris	4	100	50	50
spambase	58	4601	1813	2788
tic-tac-toe	10	958	332	626
sick	30	3772	231	3541
waveform	41	5000	1692	3308
car	6	1728	518	1210
vote	16	435	267	168
ionosphere	34	351	126	225

- The kr-vs-kp dataset represents the King Rook-King Pawn on a7 in chess. The positive class indicates a victory for white (1,669 instances, or 52%), while the negative class indicates a defeat for white (1,527 instances, 48%).
- The mushroom dataset is based on the Audubon Society Field Guide for North American Mushrooms (1981) and includes 21 attributes related to mushroom characteristics such as cap shape, surface, odor, and ring type.
- The Iris plants dataset comprises three classes, each with 50 instances, representing different iris plant species. The dataset contains four numerical attributes in centimeters: sepal length, sepal width, petal length, and petal width.
- The Spambase dataset consists of 57 attributes, each representing the frequency of a term appearing in an email, such as the “address”.
- The tic-tac-toe endgame dataset encodes all possible board configurations for the game, with “x” playing first. It contains attributes (X, O, and blank) corresponding to each of the nine tic-tac-toe squares.
- The Sick dataset includes thyroid disease records from the Garvan Institute and J. Ross Quinlan of the New South Wales Institute in Sydney, Australia. 3,772 instances with 30 attributes (seven continuous and 23 discrete) and 5.4% missing data. Attributes include age, pregnancy, TSH, T3, TT4, etc.

- The waveform dataset generator comprises 41 attributes and three wave types, with each class consisting of two “base” waves.
- The Car Evaluation Dataset rates cars on price, buying, maintenance, comfort, doors, capacity, luggage, boot size, and safety using a simple hierarchical decision model. The dataset consists of 1,728 instances categorized as unacceptable, acceptable, good, and very good.
- The 1984 US Congressional Voting Records dataset shows how members voted on 16 CQA-identified critical votes. Votes are divided into nine categories, simplified to yea, nay, or unknown disposition. The dataset has two classes: Democrats (267) and Republicans (168).
- The Johns Hopkins Ionosphere dataset contains data collected near Goose Bay, Labrador, using a phased array of 16 high-frequency antennas. “Good” radar returns show ionosphere structure, while “bad” returns are ionosphere-free. The dataset includes 351 instances with 34 attributes categorized as good or bad.

All datasets were transformed into a two-class binary problem for comparison with existing benchmarks. For instance, only the first and second classes were used in the “waveform” dataset, and the first two classes were utilized in the “Iris” dataset. We generated multiple fictitious label sets for each dataset to simulate the crowdsourcing concept of collecting several crowd labels for each instance. We selected random samples in the datasets using a uniform distribution and altered their corresponding true labels to incorrect ones, while maintaining the original distribution of the ground-truth labels. The probability of each instance containing the correct true label was determined using a uniform distribution, allowing us to create synthetic label sets for each worker that preserved the underlying structure and difficulty of the original classification problem. By creating datasets with various levels of accuracy, we can evaluate the performance of the proposed method under different conditions of worker expertise and reliability. This allows us to assess the ability of our method to handle diverse real-world crowdsourcing scenarios and gain insight into its general applicability and effectiveness in improving overall classification accuracy.

1.2 Benchmarks

Tao [7] and Sheng [6] techniques were implemented in Python to evaluate their performance. Furthermore, the crowd-kit package (A General-Purpose Crowdsourcing Computational Quality Control Toolkit for Python) [9] was used to implement the remaining benchmark techniques, including Gold Majority Voting, MMSR [5], Wawa, Zero-Based Skill, GLAD [10], and Dawid Skene [2].

- **Worker Agreement with Aggregate (WAWA) [8]:** Wawa, also referred to as 'inter-rater agreement', is a metric used in crowdsourcing jobs that do not employ test questions [app]. The WAWA algorithm consists of three steps: it calculates the majority vote label, estimates workers' skills as a fraction, and calculates the agreement between workers and the majority vote [8].
- **Zero-Based-Skill (ZBS) [8]:** employs a weighted majority vote (WMV). After processing a collection of instances, it re-evaluates the abilities of the workers based on the accuracy of their responses. This process is repeated until the labels no longer change or the maximum number of iterations is reached.
- **Karger-Oh-Shah (KOS) [8]:** Iterative algorithm that calculates the log-likelihood of the task being positive while modeling the reliabilities of the workers. Let $A_{(i,\alpha)}$ be a matrix of answers of worker α on task i. $A_{(i,\alpha)} = 0$ if worker α didn't answer the task i otherwise $|A_{(i,\alpha)}| = 1$. The algorithm operates on real-valued task messages $x_{i\rightarrow\alpha}$ and worker messages $y^{\alpha\rightarrow i}$. A task message $x_{i\rightarrow\alpha}$ represents the log-likelihood of task i being a positive task, and a worker message $y_{\alpha\rightarrow i}$ represents how reliable worker α is. On

iteration k the values are updated as follows [8]:

$$x_{i\to\alpha}^{(k)} = \sum_{\alpha'\in\partial i\backslash\alpha} A_{(i,\alpha')} y_{\alpha'\to i}^{(k-1)} y_{\alpha\to i}^{(k)} = \sum_{i'\in\partial\alpha\backslash i} A_{(i',\alpha)} x_{i'\to\alpha}^{(k-1)} \tag{10}$$

- **Multi-Annotator Competence Estimation (MACE) [4, 8]:** Probabilistic model that associates each worker with a probability distribution over the labels. For each task, a worker might be in a spamming or not spamming state. If the worker is not spamming, they will yield the correct label. If the worker is spamming, they answer according to their probability distribution. Let's assume that the correct label $y^{(i)}$ comes from a discrete uniform distribution. When a worker annotates the task, they are in the spamming state with probability Bernoulli$(1-\theta_\alpha)$. So, if their state $s_\alpha = 0$, their response is $z_\alpha^{(i)} = y^{(i)}$. Otherwise, their response $z^{(i,\alpha)}$ is drawn from a multinomial distribution with parameters ξ_α.
- **Matrix Mean-Subsequence-Reduced Algorithm (MMSR) [5, 8]:** The MMSR assumes that workers have different levels of expertise and are associated with a vector of "skills" $\boldsymbol{s}$ which has entries s_α showing the probability that the worker α answers correctly to the given task. Having that, we can show that

$$\mathbb{E}\left[\frac{K}{K-1}\widetilde{C} - \frac{1}{K-1}\mathbf{1}\mathbf{1}^T\right] = \boldsymbol{s}\boldsymbol{s}^T, \tag{11}$$

where K is the total number of classes, $\widetilde{C}$ is a covariation matrix between workers, and $\mathbf{1}\mathbf{1}^T$ is the all-ones matrix, which has the same size as $\widetilde{C}$. So, the problem of recovering the skills vector $\boldsymbol{s}$ becomes equivalent to the rank-one matrix completion problem. The MMSR algorithm is an iterative algorithm for rank-one matrix completion, so its result is an estimator of the vector $\boldsymbol{s}$. Then, the aggregation is the weighted majority vote with weights equal to $\log\frac{(K-1)s_\alpha}{1-s_\alpha}$.

- **Generative model of Labels, Abilities, and Difficulties (GLAD) [8, 10]:** A probabilistic model that parameterizes workers' abilities and tasks' difficulties. Let's consider a case of K class classification. Let p be a vector of prior class probabilities, $\omega_\alpha \in (-\infty, +\infty)$ be a worker's ability parameter, $\beta^{(k)} \in (0, +\infty)$ be an inverse task's difficulty, $y^{(k)}$ be a latent variable representing the true task's label, and $z_\alpha^{(k)}$ be a worker's response that we observe. The relationships between these variables and parameters according to GLAD are represented by the following latent label model. The prior probability of $y^{(k)}$ being equal to c is $\Pr(y^{(k)} = c) = p[c]$, the probability distribution of the worker's responses conditioned by the true label value c follows the single coin Dawid-Skene model, where the true label probability is a sigmoid function of the product of the worker's ability and the inverse task's difficulty:

$$\Pr\left(z_\alpha^{(k)} = j \,|\, y^{(k)} = c\right) = \begin{cases} f(\alpha, k), & j = c \\ \frac{1 - f(\alpha, k)}{K - 1}, & j \neq c \end{cases} \tag{12}$$

  where $f(\alpha, k) = \frac{1}{1 + e^{-\omega_\alpha \beta^{(k)}}}$. Parameters p, ω, β and latent variables y are optimized through the expectation-minimization algorithm.
- **Dawid-Skene [2, 8]:** Probabilistic model that parameterizes workers' level of expertise through confusion matrices. Let e^α be a worker's confusion (error) matrix of size $K \times K$ in case of K class classification, p be a vector of prior class probabilities, $y^{(i)}$ be a true task's label, and $z_\alpha^{(i)}$ be a worker's answer for the task i. The relationships between these parameters are represented by the following latent label model. Here, the prior true label probability is $\Pr(y^{(i)} = c) = p[c]$ and the distribution of the worker's responses given the true label c is represented by the corresponding column of the error matrix: $\Pr(z_\alpha^{(i)} = k | y^{(i)} = c) = e^\alpha[k, c]$. Parameters p and e^α and latent variables z are optimized through the expectation-maximization algorithm.

1.3 Weight Measurement Evaluation

Following the generation of multi-label sets, the aggregate labels were determined using the proposed Crowd-Certain as well as various established methods. We examined two strategies for classifier selection, as detailed in Section ??. Because there was no substantial variation in the final outcomes observed, the second strategy was adopted for its utilization of the random forest classification technique. This choice not only conserved processing time but also decreased the need for numerous Python package dependencies. For each worker α, we trained ten distinct random forests, each comprising four trees with a maximum depth of four, under various random states, as outlined in Section ??. Figure 1 depicts the relationship between the randomly assigned workers' probability threshold ($\pi_{\alpha}^{(k)}$) and their corresponding estimated weights ($\omega_{\alpha}^{(k)}$). In Tao's method scenario, the figure presents the average weights over all instances. Notably, as the reliability (probability threshold) of a worker exceeds a particular threshold, the weight computed by Tao's method reaches a saturation point, while the proposed technique exhibits a considerably stronger correlation. The individual data points symbolize the actual calculated weights, and the curve illustrates the regression line.

1.4 Label Aggregation Evaluation

The Figure 2 portrays the accuracy comparison of our label aggregation technique, termed Crowd-Certain, against ten existing methods, evaluated over ten distinct datasets. Each dataset was labeled by three different workers, with labels generated based on a uniform distribution and specific probability thresholds Π_{α} as explained in Section ??. For a comprehensive evaluation, all experiments were repeated three times using different random seed numbers to account for randomness. The accuracy scores presented in the figure represent the average of these three runs and illustrate the degree of concordance between the aggregated label $\nu^{(i,k)}$ from each technique and the actual ground truth $y^{(i,k)}$. It is important to note that, in the execution of our proposed technique,

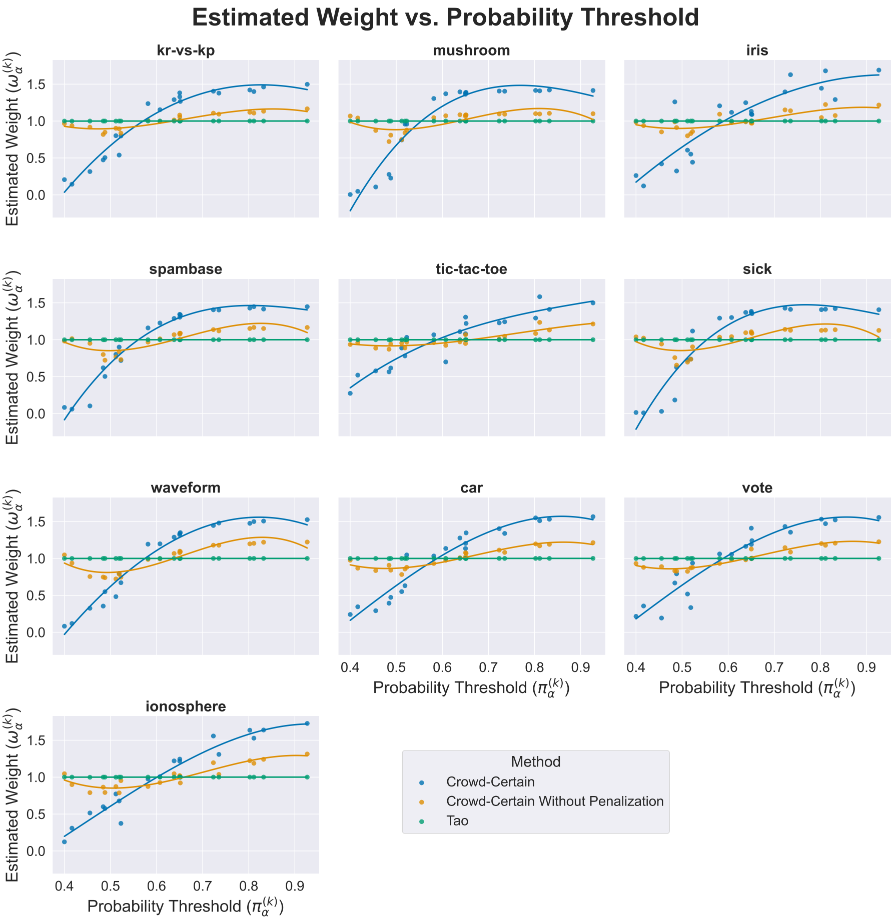

Fig. 1 A comprehensive comparison of weight computation techniques across ten distinctive datasets. Each subplot corresponds to a specific dataset, visually illustrating the relationship between the randomly assigned worker's probability threshold ($\pi_\alpha^{(k)}$) (represented on the horizontal axis) and the resulting computed weights ($\omega_\alpha^{(k)}$) (shown on the vertical axis). This relationship is analyzed for the Crowd-Certain method in two scenarios — with and without penalization — and is also compared with the Tao method [7]. Individual data points represent real measured weights, while the overlaid curve delineates the corresponding regression line.

Crowd-Certain, the aggregated labels were derived through the application of the predicted probabilities, denoted as $\eta_\alpha^{(i,k)}$. This approach is significant as it enables the reuse of trained classifiers on future sample data, eliminating the need for recurrent simulation processes — a substantial advantage in terms of computational efficiency. Conversely, the methodologies of existing techniques necessitated the use of actual crowd labels $z_\alpha^{(i,k)}$ to determine the aggregated labels. For example, in the case of Tao [7] the aggregated labels were obtained using the following equation:

$$\nu^{(i,k)} = \begin{cases} 1 & \text{if } \left(\sum_{\alpha=1}^{M} \omega_{\alpha}^{(k)} \, z_{\alpha}^{(i,k)} \right) > 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, k \tag{13}$$

These methods inherently involve re-running simulations for every new dataset, which could be computationally expensive and time-consuming. The Crowd-Certain method outperforms the existing methods, yielding higher average accuracy rates across 9 of the 10 evaluated datasets, while achieving a smaller accuracy compared to only one of the 10 benchmarks (MMSR) on tic-tac-toe dataset. For example, in the 'kr-vs-kp' dataset, our proposed Crowd-Certain method achieved an average accuracy of approximately 0.923, significantly exceeding the highest-performing existing method that reached an accuracy of about 0.784. This trend holds true across other datasets as well, such as 'mushroom', 'spambase', and 'waveform', where the Crowd-Certain method achieves superior average accuracies of around 0.98, 0.90, and 0.92, respectively. We further extended our experiment to explore the effects of varying the number of workers, ranging from 3 up to 7. The results shown in Figure 3 are presented as a series of box plots, each illustrating the distribution of accuracy (1st column), F1 (2nd column), and AUC (3rd column) scores across the 10 datasets for a given number of workers. These plots provide a visual summary of our technique's performance across various settings, including the median, quartiles, and potential outliers in the distribution of accuracies. Notably, our proposed Crowd-Certain technique shows improvements over the 10 benchmark methods for different number of workers. This enhancement is evident irrespective of the number of workers involved.

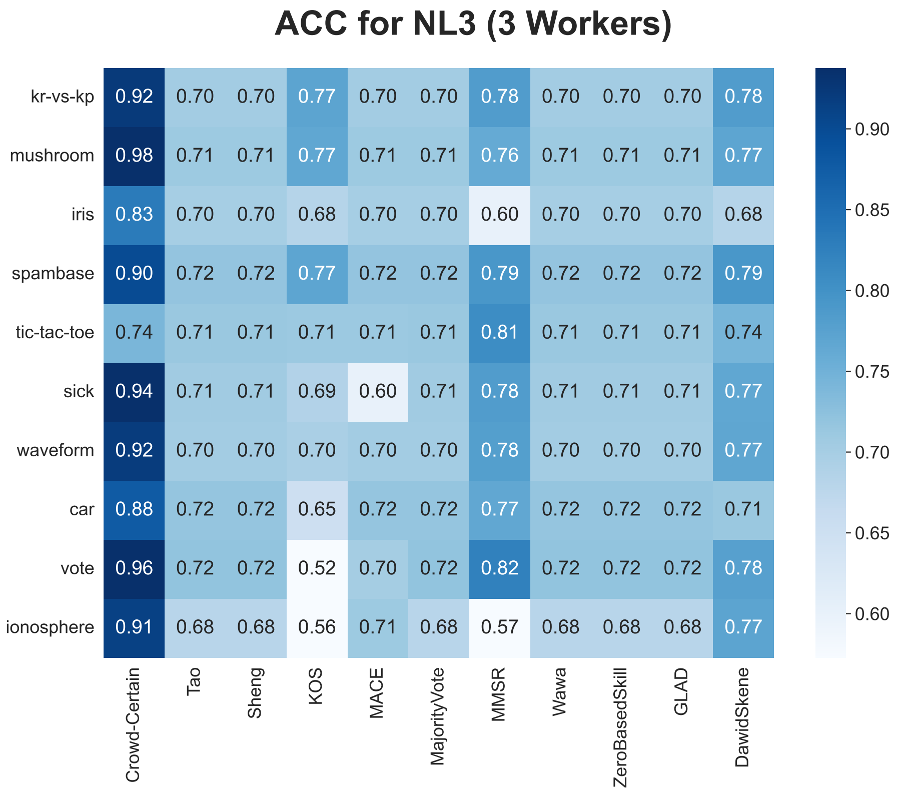

Fig. 2 Comparative analysis of the mean accuracy between the proposed Crowd-Certain method and ten pre-existing label aggregation techniques, under conditions featuring three crowd workers. The depicted mean accuracy score is derived from an averaging process across three separate trials, each initiated with a distinct random seed, thus ensuring a fair and balanced comparison. Darker blue means higher mean accuracy.

1.5 Confidence Score Evaluation

The Figure 4 presents the evaluation of the two confidence score measurement techniques, namely Freq and Beta, using two performance metrics: Expected Calibration Error (ECE) and Brier Score. The evaluations were conducted across a variety of datasets and using three techniques: Crowd-Certain, Tao, and Sheng, when using three workers. Figure 4 depicts the performance of three different strategies: Crowd-Certain, Tao, and Sheng, compared across two metrics: ECE and Brier Score. These results are obtained using two different confidence score calculation techniques, Freq and Beta, applied over ten different datasets when three workers. The ECE offers an aggregated measure of the

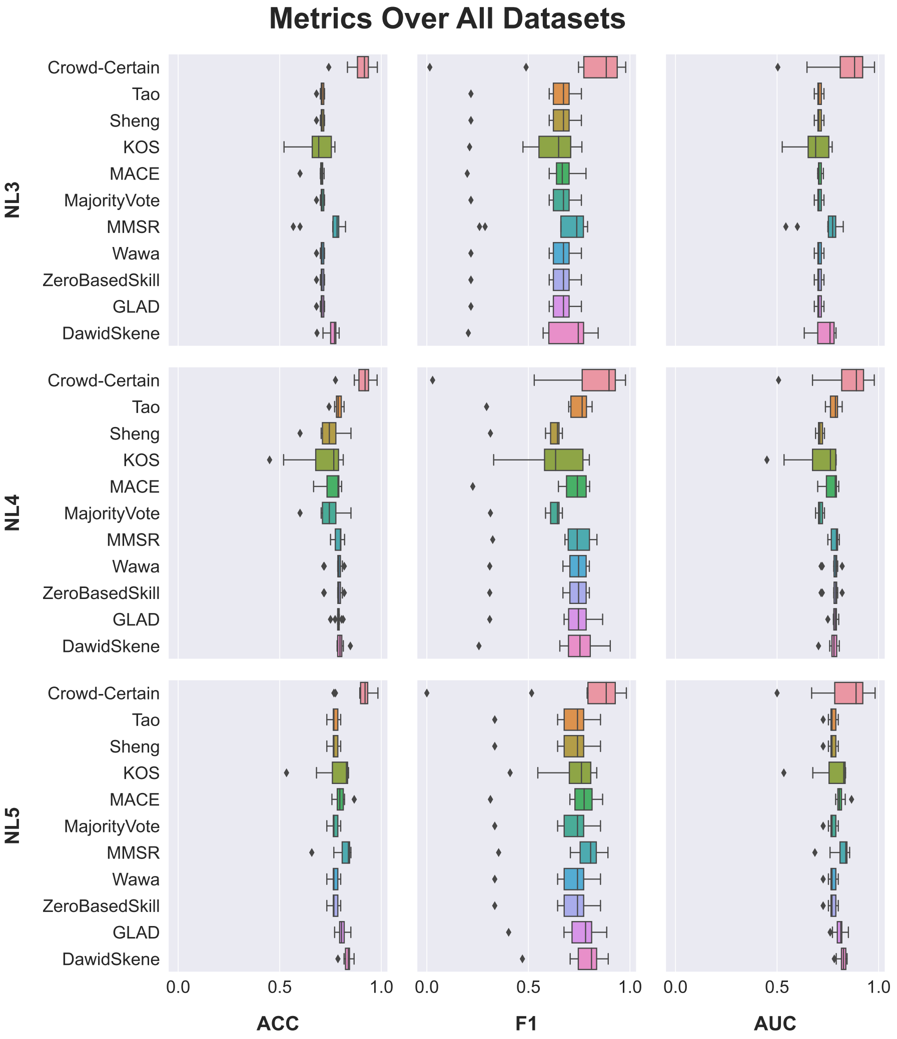

Fig. 3 This 3 × 3 structured figure provides a comprehensive comparison of Accuracy (first column), F1 (second column), and AUC scores (third column) across multiple label aggregation techniques (each shown with a unique color), including the proposed Crowd-Certain method and nine pre-existing techniques. Each row illustrates the results for a different number of crowd workers: the top row for three workers, the middle row for four workers, and the bottom row for five workers. Each subfigure presents ten boxplots, where each boxplot represents an aggregation technique. The metrics for each boxplot are computed from ten average scores, each corresponding to a distinct dataset. The average scores are derived from three independent trials, each with a different random seed. The aggregation of labels used in each experiment to calculate these metrics was obtained using Eq. (**??**) for Crowd-Certain and Eq. (13) for pre-existing techniques.

reliability of confidence scores ($F_{\Omega}^{(i,k)}$ and $F_{\beta}^{(i,k)}$) to assess the calibration of the aggregated labels across different techniques and strategies. A lower ECE indicates better-calibrated predictions, i.e., the estimated confidence scores are closer to the ground truth labels.

Brier Score is a metric that quantifies the accuracy of probabilistic predictions. It calculates the mean squared difference between the estimated confidence scores ($F_{\Omega}^{(i,k)}$ and $F_{\beta}^{(i,k)}$) and the ground truth labels ($y^{(i,k)}$). Hence, higher Brier Score values correspond to better model performance. In Figure 4, it can be observed that for the Brier Score metric for both Beta (F_{β}) and Freq (F_{Ω}) strategies, across all datasets, the proposed Crowd-Certain strategy consistently achieves higher scores when compared to Tao and Sheng. This indicates that the Crowd-Certain strategy offers better-calibrated predictions, providing a higher level of confidence in the aggregated labels. For the ECE metric and Beta strategy (F_{β}) the Crowd-Certain strategy outperforms Tao and Sheng across most datasets. For the ECE metric and Freq strategy (F_{Ω}), the Tao technique generally results in higher ECE, indicating worse calibration, whereas the Crowd-Certain and Sheng techniques show varying performance depending on the number of workers. Figure 5 showcases the results for two metrics, ECE and Brier Score, for two confidence measurement techniques (Beta (F_{β}) and Freq (F_{Ω}) strategies), applied using three different techniques: Crowd-Certain, Tao, and Sheng. These results are obtained for the kr-vs-kp dataset under different numbers of workers from 3 (denoted with NL3) up to (denoted with NL7). In general, the Brier Score decreases and ECE increases as the number of workers increases, which suggests that increasing the number of workers does not necessarily improve the performance. The performance varies depending on the confidence measurement technique and the strategy used. For the Freq strategy, the Crowd-Certain technique yields lower ECE and higher Brier Score across nearly all numbers of workers compared to the Tao and Sheng techniques, indicating more confident predictions when having only 3 workers. For the Beta strategy, the performance varies between techniques. For the

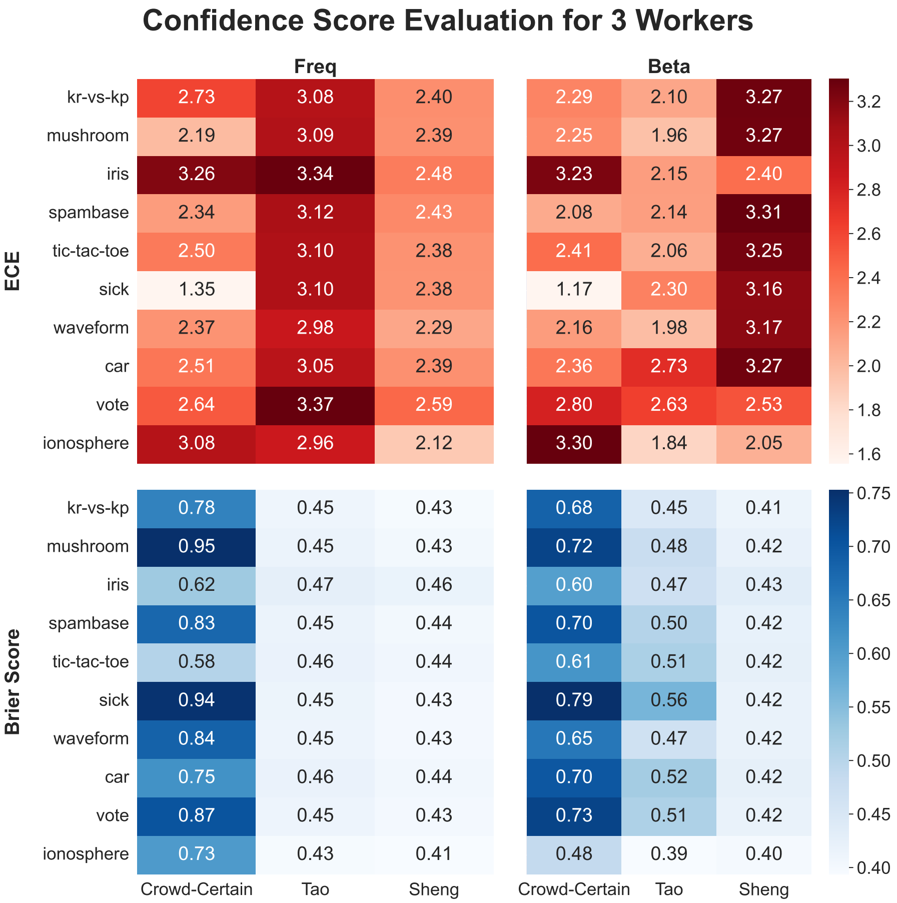

Fig. 4 Comparative heatmap of the ECE and Brier Score across two confidence score measurement strategies: Beta (F_β) and Freq (F_Ω). The comparison involves three different label aggregation techniques: Crowd-Certain, Tao, and Sheng, and spans ten distinct datasets for three crowd workers (NL3). The chosen metrics provide insight into the calibration of the predictions across different configurations

Brier Score, the Freq strategy combined with the Crowd-Certain technique performs better across all numbers of workers compared to other combinations of techniques and strategies. For the ECE, the Beta strategy combined with the Crowd-Certain technique yields the lowest values for three and four workers, indicating a good match between predicted confidences and observed frequencies. However, the ECE generally exhibits a tendency to increase as the number of workers increases, indicating a decline in calibration. Overall, these results suggest that the choice of the confidence measurement technique

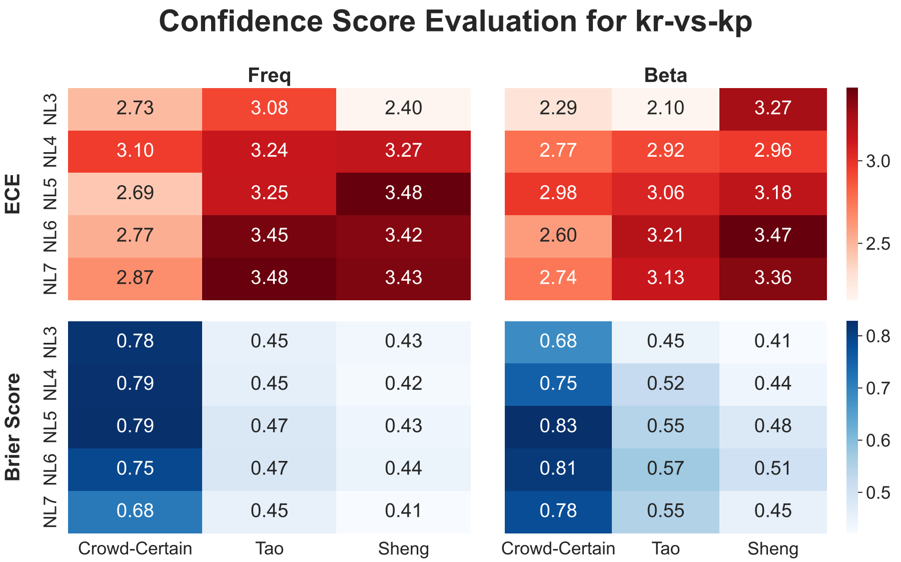

Fig. 5 Comparative evaluation of the ECE and the Brier Score across two confidence score measurement strategies: Beta (F_β) and Freq (F_Ω). The results are plotted for three distinct label aggregation techniques — Crowd-Certain, Tao, and Sheng —– on the kr-vs-kp dataset with varying numbers of crowd workers from three to seven (NL3 to NL7).

and the strategy have significant impacts on the calibration (confidence) of the predictions. Further investigations could be beneficial to understand the specific conditions under which certain techniques and strategies yield superior performance.

2 Discussion

Label aggregation is a critical component of crowdsourcing and ensemble learning strategies. Many generic label aggregation algorithms fall short because they do not account for the varying reliability of the workers. In this work, we introduced a new method for crowd labeling aggregation termed as Crowd-Certain. This technique effectively leverages uncertainty measurements to refine the aggregation of labels obtained from multiple workers. Through an extensive comparative analysis, it was shown to yield higher accuracy in label aggregation against ground truth, particularly in settings where only a limited number of workers are available. This advantage over established methods such as Gold Majority Vote, MV, MMSR, Wawa, Zero-Based Skill, GLAD, and Dawid Skene demonstrates the potential of the proposed method in enhancing the reliability of label aggregation in crowdsourcing and ensemble learning applications. Our approach is distinguished by its application of a weighted soft majority voting scheme, where the weights are determined based on the level of uncertainty associated with each worker's labels. Importantly, the proposed technique takes into account the possibility of consistently inaccurate workers and includes measures to penalize them (shown in Eq. **??**), thus ensuring the credibility of the computed weights $\omega_{\alpha}^{(k)}$. The calculated weights follow a pre-set ground-truth accuracy closely, highlighting the effectiveness of the technique in capturing the quality of workers' labels. Moreover, the Crowd-Certain technique demonstrates an appreciable capability to generate confidence scores that accompany each aggregated label, offering an extended context that can be invaluable in practical applications. In this study, we evaluated various techniques for aggregating crowdsourced labels and measuring the confidence scores associated with these labels. This evaluation involved two key metrics (ECE and Brier Score) for the evaluation of confidence scores, as well as three metrics (accuracy, AUC, and F1 score) for the evaluation of the aggregated labels (ν). These metrics assessed different facets of model performance: calibration of the confidence scores (how confident the predictions are), and the performance of the aggregated labels against the ground truth. By comparison to existing methodologies, our method demonstrates superior performance across a variety of datasets, yielding higher average accuracy rates. Furthermore, our experiments, which involved varying the number of workers, demonstrated that Crowd-Certain outperforms the benchmark methods in nearly all scenarios, irrespective of the number of workers involved. This indicates that our approach is not only accurate and efficient, but also highly versatile to a range of settings, showing consistent improvements across different numbers of workers. Significantly, our technique introduces an advantageous property by assigning a single weight ($\omega_{\alpha}^{(k)}$ for class k) to each worker

α for all instances in the dataset. Moreover, the application of predicted probabilities ($\eta_{\alpha}^{(i,k)}$) in our method allows for the reuse of trained classifiers on future sample data, which eliminates the need for recurrent simulation processes. This presents a distinct advantage over conventional techniques, which require computationally expensive and time-consuming repeated simulations for every new dataset. It's worth noting that Crowd-Certain outperforms in nearly all evaluated scenarios across the tested datasets with one exception where the accuracy is lower than MMSR technique on tic-tac-toe dataset as shown in Figures 2 and 3. This consistency is evident even when considering variance in dataset characteristics, such as 'kr-vs-kp', 'mushroom', and 'spambase'. In addition to label aggregation, the evaluation of confidence score measurements revealed further advantages of the Crowd-Certain method. When analyzing two confidence score measurement techniques, Freq and Beta, we found that our strategy achieves lower ECE scores compared to Tao and Sheng for most datasets. This implies that Crowd-Certain provides better-calibrated predictions, offering a higher level of confidence in the aggregated labels. Furthermore, Crowd-Certain also outperformed other techniques in terms of Brier Score across all datasets, indicating a higher accuracy of probabilistic predictions. Our results indicate that the choice of aggregation and confidence measurement technique can significantly impact the performance. Furthermore, it shows that increasing the number of workers does not necessarily improve the performance, as indicated by the general increase in ECE and decrease (for Freq strategy) in Brier Score with a higher number of workers. This suggests a trade-off between the number of workers and the performance, and that the optimal number may depend on the specific context and the chosen techniques.

3 Conclusion

The proposed Crowd-Certain label aggregation technique offers a promising solution for crowdsourced labeling tasks by providing a superior accuracy across various settings. Furthermore, it improves computational efficiency by allowing for the reuse of trained classifiers on future sample data, making it a viable option for large-scale data labeling tasks. While our findings are encouraging, further research and validation across more diverse datasets and real-world scenarios are warranted to further refine and enhance this approach. Future work could delve deeper into understanding why certain techniques and strategies outperform others under specific conditions. Further investigations could explore the effects of other factors such as the complexity of the task and the diversity of the crowd, which may impact the performance of different techniques and strategies. Our findings could guide future research and applications in this domain, with potential implications for various fields that rely on crowdsourced data, including machine learning, data science, and citizen science.

4 Availability of Data and Materials

The source code can be found at https://github.com/artinmajdi/crowdcertain

Competing Interests

The authors declare that they have no competing interests.

References

[app] Calculating Worker Agreement with Aggregate (Wawa).

[2] Dawid, A. P. and Skene, A. M. (1979). Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied Statistics*, 28(1):20.

[3] Duan, D. and Graff, C. (2017). UCI Machine Learning Repository.

[4] Hovy, D., Berg-Kirkpatrick, T., Vaswani, A., and Hovy, E. (2013). Learning Whom to Trust with MACE. In *North American Chapter of the Association for Computational Linguistics*.

[5] Ma, Q. and Olshevsky, A. (2020). Adversarial Crowdsourcing Through Robust Rank-One Matrix Completion. In *Adv. Neural Inf. Process. Syst.*, volume 33, pages 21841–21852. Curran Associates, Inc.

[6] Sheng, V. S., Zhang, J., Gu, B., and Wu, X. (2019). Majority Voting and Pairing With Multiple Noisy Labeling. *IEEE Trans. Knowl. Data Eng.*, 31(7):1355–1368.

[7] Tao, F., Jiang, L., and Li, C. (2020). Label Similarity-Based Weighted Soft Majority Voting and Pairing for Crowdsourcing. *Knowl Inf Syst*, 62(7):2521–2538.

[8] Toloka-AI (2023). Crowd-Kit Documentation. Toloka AI.

[9] Ustalov, D., Pavlichenko, N., and Tseitlin, B. (2021). Learning From Crowds With Crowd-Kit.

[10] Whitehill, J., Wu, T.-f., Bergsma, J., Movellan, J., and Ruvolo, P. (2009). Whose Vote Should Count More: Optimal Integration of Labels From Labelers of Unknown Expertise. In *Adv. Neural Inf. Process. Syst.*, volume 22. Curran Associates, Inc.