DATA AND DOMAIN UNCERTAINTY IN MACHINE LEARNING

by

Artin Majdi

A Dissertation Submitted to the Faculty of the
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of
DOCTOR OF PHILOSOPHY

In the Graduate College
UNIVERSITY OF ARIZONA

2023

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Artin Majdi, titled *DATA AND DOMAIN UNCERTAINTY IN MACHINE LEARNING*, and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Jeffrey J. Rodriguez	Date
Carlos Alsua	Date
Ali Bilgin	Date
Greg Ditzler	Date

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Jeffrey J. Rodriguez Dissertation Director	Date

STATEMENT by AUTHOR

This dissertation *DATA AND DOMAIN UNCERTAINTY IN MACHINE LEARNING* prepared by Artin Majdi has been submitted in partial fulfillment of requirements for a doctoral degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under the rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by

SIGNED: _____

ACKNOWLEDGMENTS

As I reflect upon the completion of this invigorating journey, my heart brims with profound gratitude towards those who journeyed with me, imparting their wisdom, support, and guidance at every step.

First and foremost, my deepest appreciation is directed towards my mentor, Dr. Jeffrey J. Rodriguez, whose steadfast guidance and support throughout the project has been unparalleled. His expertise, dedication, and continual encouragement have been priceless to me, for which I am eternally thankful.

My sincere gratitude goes out to Mr. Nirav Merchant, Ms. Maliaca Oxnam, and Drs. Carlos Alsua, Manoj Saranathan, and Mahesh Keerthivasan, whose unceasing support has been invaluable throughout this endeavor. Additionally, I wish to convey my gratitude to Drs. Ali Bilgin and Abhijit Mahalanobis for their insightful contributions and feedback which significantly enhanced the depth and quality of my work. Your invaluable support and mentorship have played a crucial role in my personal and academic evolution.

An acknowledgment wouldn't be complete without a heartfelt tribute to my family. Your unwavering faith in my capabilities, your patience during the tough times, and your consistent encouragement have served as my strongest pillars of support.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Crowd-Certain: Uncertainty-Based Weighted Soft Majority Voting with Applications in Crowdsourcing and Ensemble Learning

Crowdsourcing systems have been used to accumulate massive amounts of labeled data for applications such as computer vision and natural language processing. However, because crowdsourced labeling is inherently dynamic and uncertain, developing a technique that can work in most situations is extremely challenging. In this paper, we propose a novel method called "crowd-certain", which provides a more accurate and reliable aggregation of labels, ultimately leading to improved overall performance in both crowdsourcing and ensemble learning scenarios. The proposed method uses the consistency and accuracy of the annotators as a measure of their reliability relative to other annotators. The experimental results show that the proposed technique generates a weight that closely follows the annotator's degree of reliability. Moreover, the proposed method uses the consistency and accuracy of annotators as a measure of their reliability relative to other annotators. Experiments performed on a variety of crowdsourcing datasets indicate that the proposed method outperforms prior methods in terms of accuracy, with significant improvement over all investigated benchmarks (Gold Majority Vote, MV, MMSR, Wawa, Zero-Based Skill, GLAD, and Dawid Skene), particularly when few annotators are available. **KEYWORDS:** Supervised learning, crowdsourcing, confidence score, soft weighted majority voting, label aggregation, annotator quality, error rate estimation, multi-class classification, ensemble learning, uncertainty measurement

1.1 Introduction

Supervised learning techniques require a large amount of labeled data to train models to classify new data [2, 3]. Traditionally, data labeling has been assigned to experts in the domain or well-trained annotators [4]. Although this method produces high-quality labels, it is inefficient and costly [5, 6]. Social networking provides an innovative solution to the labeling problem by allowing data to be labeled by online crowd annotators. This has become feasible, as crowdsourcing services such as Amazon Mechanical Turk (formerly CrowdFlower) have grown in popularity. Crowdsourcing systems have been used to accumulate large amounts of labeled data for applications such as computer vision [7, 8] and natural language processing [9]. However, because of individual differences in preferences and cognitive abilities, the quality of labels acquired by a single crowd annotator is typically low, thus jeopardizing applications that rely on these data. This is because crowd annotators are not necessarily domain experts and may lack the necessary training or expertise to produce high-quality labels. Aggregation after repeated labeling is one method for handling annotators with various abilities. Label aggregation is a process used to infer an aggregated label for a data instance from a multi-label set [10]. Several studies have demonstrated the efficacy of repeated labeling [11, 12]. Repeat labeling is a technique in which the same data are labeled by multiple annotators, and the results are combined to estimate an aggregated label using majority voting (MV) or other techniques. In the case of MV, an aggregated label is the label that receives the most votes from the annotators for a given data instance. This can help reduce the impact of biases or inconsistencies made by annotators. Several factors, such as problem-specific characteristics, the quality of the labels created by the annotators, and the amount of data available, can influence the effectiveness of the aggregation methodologies. Consequently, it is difficult to identify a clear winner among the different techniques. For example, in binary labeling, one study [10] discovered that Raykar's [13] technique outperformed other aggregation techniques. However, according to another study [14], the traditional Dawid-Skene (DS) model [15] was more reliable in multi-class settings (where data instances can be labeled as belonging to multiple classes). Furthermore, regardless of the aggregation technique used, the performance of many aggregation techniques in real-world datasets remains unsatisfactory [16]. This can be attributed to the complexity of these datasets, which often do not align with the assumptions and limitations of different methods. For example, real-world datasets may present issues such as labeling inaccuracies, class imbalances, or overwhelming sizes that challenge efficient processing with available resources. These factors can adversely affect the effectiveness of label aggregation techniques, potentially yielding less than optimal results for real-world datasets. Prior information may be used to enhance the label aggregation procedure. This can include domain knowledge, the use of quality control measures and techniques

that account for the unique characteristics of annotators and data. Knowing the reliability of certain annotators, it is possible to draw more accurate conclusions about labels [17]. For instance, in the label aggregation process, labels produced by more reliable annotators (such as domain experts) may be given greater weight. The results of the label aggregation process can also be validated using expert input [18]. During the labeling process, domain experts can provide valuable guidance and oversight to ensure that the labels produced are accurate and consistent. The agnostic requirement for general-purpose label aggregation is that label aggregation cannot use information outside the labels themselves. This requirement is not satisfied in most label aggregation techniques [19]. The agnostic requirement ensures that the label aggregation technique is as general as possible and applicable to a wide range of domains with minimal or no additional context. The uncertainty of annotators during labeling can provide valuable prior knowledge to determine the appropriate amount of confidence to grant each annotator while still adhering to the requirement of a general-purpose label aggregation technique. We developed a method for estimating the reliability of different annotators based on the annotator's own consistency during labeling. We take this concept a step further by calculating a weight for each annotator based not only on their own reliability but also on the reliability scores of all other workers involved. This consideration of inter-reliability ensures a more comprehensive and dynamic weighting process, adjusting to the overall performance of the entire group of annotators. Thus, the generated weights become a robust measure of both individual and collective trustworthiness, which significantly improves the accuracy and efficacy of our labeling aggregation method. We propose a novel method called "crowd-certain", which provides a more accurate aggregation of labels, ultimately leading to improved overall performance in both crowdsourcing and ensemble learning scenarios. The proposed method uses the consistency of annotators versus a trained classifier to determine their reliability. The experimental results conducted on multiple crowdsourcing datasets show that the proposed techniques generate a weight that closely follows a pre-set ground-truth accuracy, for that each annotator. The proposed techniques outperform prior methods (Gold Majority Vote, MV, MMSR, Wawa, Zero-Based Skill, GLAD, and Dawid Skene), in terms of accuracy of the aggregated labels with respect to the ground truth labels, particularly when few annotators are available. The remainder of this paper is organized as follows. Section 1.2 examines related work involving label aggregation algorithms. In Section 1.3, we provide a detailed explanation of our proposed technique. Section1.4 presents the experiments and findings. Finally, Section 1.5 summarizes the findings and identifies the main directions for future research.

1.2 Related Work

Numerous label aggregation algorithms have been developed to capture the complexity of crowdsourced labeling systems, including techniques based on annotator reliability [20,21], confusion matrices [13,22], intentions [20, 23], biases [24, 25, 26], and correlations [27]. However, because crowdsourced labeling is inherently dynamic and uncertain, developing a technique that can work in most situations is extremely challenging. Many techniques [8, 9, 13, 28, 29] utilize the Dawid and Skene (DS) generative model [15]. Ghosh [29] extended the DS model by using singular value decomposition (SVD) to calculate the reliability of the annotator. Similarly to Ghosh [29], Dalvi [28] used SVD to estimate true labels with a focus on the sparsity of the labeling matrix. In crowdsourcing, it is common for the labeling matrix to be sparse, meaning that not all annotators have labeled all the data. This may be due to several factors, such as the cost of labeling all data instances or the annotators' time constraints. Karger [9] described an iterative strategy for binary labeling based on a one-coin model [29]. Karger [9] extends the one-coin model to multi-class labeling by converting the problem into $k - 1$ binary problems (solved iteratively), where k is the number of classes. The MV technique assumes that all annotators are equally reliable. For segmentation, Warfield [30] proposed simultaneous truth and performance level estimation (STAPLE), a label fusion method based on expectation maximization. STAPLE "weighs" expert opinions during label aggregation by modeling their reliability. Since then, many variants of this technique have been proposed [31, 32, 33, 34, 35, 36, 37, 38]. The problem with these label aggregation approaches is that they require the computation of a unique set of weights for each sample, necessitating the re-evaluation of the annotators' weights when a new instance is added. Among the numerous existing label aggregation strategies, MV remains the most efficient and widely used approach [1]. If we assume that all annotators are equally reliable and that their errors are independent of one another, then, according to the theory of large numbers, the likelihood that the MV is accurate increases as the number of annotators increases. However, the assumption that all annotators are equally competent and independent may not always hold. Furthermore, MV does not provide any additional information on the degree of disagreement among the annotators (As an example, consider the scenario where four of seven doctors think patient A needs immediate surgery, while all seven think patient B needs immediate surgery; MV will simply label "yes" in both cases). To address this problem, additional measures such as inter-annotator agreement (IAA) have been used [39]. IAA is a measurement of the agreement among multiple annotators who label the same data instance. Typically, IAA is calculated using statistical measures, such as Cohen's kappa, Fleiss's kappa, or Krippendorff's alpha [40]. These measures consider both the observed agreement between the annotators and the expected agreement owing to random chance. IAA can also be visualized using a confusion matrix or

annotation heatmap, which illustrates the distribution of labels assigned by the annotators. This can help identify instances where the annotators disagree or are uncertain and can guide further analysis to improve the annotation [41]. Recently, Sheng [42] proposed a technique that provided a confidence score along with an aggregated label. The main problem with this approach is that it assumes that all annotators are equally capable when calculating the confidence score. Tao [1] improved Sheng's approach by assigning different weights to annotators for each instance. This weighting method combines the specific quality $s_{\alpha}^{(i,k)}$ for the annotator α and instance i and the overall quality τ_{α} across all instances. Inspired by Li's technique [43], Tao evaluates the similarity between the annotator labels for each instance. To derive the specific quality $s_{\alpha}^{(i)}$, Tao counts the number of annotators who assigned the same label as the annotator α for that instance. To calculate the overall quality τ_{α}, Tao performs a 10-fold cross-validation to train each of the 10 classifiers on a different subset of data using the labels provided by the annotator α as true labels and then assigns the average accuracy of the classifiers across all remaining instances as τ_{α}. The final weight for annotator α and instance i is then calculated using the sigmoid function $\gamma_{i,\alpha} = \tau_{\alpha}\left(1 + \left(s_{\alpha}^{(i)}\right)^2\right)$. However, Tao's technique [1] has some drawbacks. It relies on the labels of other annotators to estimate $s_{\alpha}^{(i)}$. However, different annotators have varying levels of competence (reliability) when labeling the data, and therefore, relying on their labels to measure $s_{\alpha}^{(i)}$ will result in propagating the errors and biases of their labels during weight estimation. Furthermore, Tao's technique [1] relies on the labels provided by each annotator α to estimate their respective τ_{α} by assuming that the trained classifiers can learn the inherent characteristics of the datasets even in the absence of ground truth labels. While that may be true in some cases, it typically leads to suboptimal measurement and the propagation of biases and errors, from both the annotator's labels and the classifier, into weight estimation.

1.3 Methods

We propose a novel method called "crowd-certain" which focuses on leveraging uncertainty measurements to improve decision-making in crowdsourcing and ensemble learning scenarios. Crowd-Certain employs a weighted soft majority voting approach, where the weights are determined based on the uncertainty associated with each annotator's labels. Initially, we use uncertainty measurement techniques to calculate the degree of consistency of each annotator during labeling. Furthermore, to ensure that the proposed technique does not calculate a high weight for annotators who are consistently wrong (for example, when a specific annotator always mislabels a specific class, and hence demonstrates a high consistency even if they label instances incorrectly), we extend the proposed technique by penalizing the annotators for instances in which they disagree with the aggregated label obtained using MV.

To mitigate the reliance on training a classifier on an annotator's labels, which may be inaccurate, we train an ensemble of classifiers for each annotator. In addition, we report two confidence scores along with the aggregated label to provide additional context for each calculated aggregate label. We report a single weight for all instances in the dataset. As will be demonstrated in Section 1.4, the proposed crowd-certain method is not only comparable to other techniques in terms of accuracy of the aggregated labels with respect to the ground truth labels for scenarios with a large number of annotators, but also provides a significant improvement in accuracy for scenarios where the number of annotators may be limited. Furthermore, by assigning a single weight to each annotator for all instances in the dataset, the model can assign labels to new test instances without recalculating the annotator weights. This is especially advantageous in situations where annotators are scarce as it enables the model to make accurate predictions with minimal dependence on the annotator input. This characteristic of the crowd-certain method can significantly reduce the time and resources required for labeling in practical applications. When deploying the model in real-world scenarios such as medical diagnosis, fraud detection, or sentiment analysis, it could be advantageous to be able to assign labels to new instances without constantly recalculating annotator weights.

1.3.1 Glossary of Symbols

For convenience, the following list summarizes the major symbols used in the subsequent discussion:

N: Number of instances.

M: Number of annotators.

$y^{(i,k)} \in \{0, 1\}$: True label for the k-th class for instance i.

$z_\alpha^{(i,k)} \in \{0, 1\}$: Label given by annotator α for k-th class for instance i.

$\underset{\alpha}{\mathrm{MV}}\left(z_\alpha^{(i,k)}\right)$: Majority voting technique (the label that receives the most votes) applied to annotator labels for class k and instance i.

$\pi_\alpha^{(k)}$: Probability threshold used as pre-set ground truth accuracy, for each annotator α and class k. It is used to generate sample binary labels (fictitious ground truth label set) for annotator α for class k. For example, the threshold values may be obtained from a uniform distribution in the

interval 0.4 to 1, i.e., $\pi_{\alpha}^{(k)} \sim U(0.4, 1)$.

$X^{(i)}$: Data for instance i.

$Y^{(i)} = \left\{y^{(i,1)}, y^{(i,2)}, \ldots, y^{(i,K)}\right\}$: True label set, for instance i. For example, consider a dataset that is labeled for the presence of cats, dogs, and rabbits in any given instance. If a given instance $X^{(i)}$ has cats and dogs but not rabbits, then $Y^{(i)} = \{1, 1, 0\}$.

$Z_{\alpha}^{(i)} = \left\{z_{\alpha}^{(i,1)}, z_{\alpha}^{(i,2)}, \ldots, z_{\alpha}^{(i,K)}\right\}$: Label set given by the annotator α for instance i.

K: number of categories (aka classes) in a multi-class multi-label problem. For example, if we have a dataset labeled for the presence of cats, dogs, and rabbits in any given instance, then $K = 3$.

$\rho^{(i)}$: Randomly generated number between 0 and 1 for instance i. It is obtained from a uniform distribution, i.e., $\rho^{(i)} \sim U(0, 1)$ This number is used to determine, for each instance i, whether the true label should be assigned to each fictitious annotator's label. For each class k, if the annotator's probability threshold $\pi_{\alpha}^{(k)}$ is greater than $\rho^{(i)}$, the true label $y^{(i,k)}$ is assigned; otherwise, an incorrect label $1 - y^{(i,k)}$ is assigned.

$\Pi_{\alpha} = \left\{\pi_{\alpha}^{(1)}, \pi_{\alpha}^{(2)}, \ldots, \pi_{\alpha}^{(K)}\right\}$: set of K probability thresholds for annotator α.

$\mathbb{X} = \left\{X^{(i)}\right\}_{i=1}^{N}$: Set of all instances.

$\mathbb{Y} = \left\{Y^{(i)}\right\}_{i=1}^{N}$: Set of all true labels.

$\mathbb{Z}_{\alpha} = \left\{Z_{\alpha}^{(i)}\right\}_{i=1}^{N}$: Set of all labels for the annotator α.

$\widehat{\mathbb{Y}} = \left\{\widehat{Y}^{(i)}\right\}_{i=1}^{N}$: Set of all aggregated labels.

$\mathbb{P} = \left\{\rho^{(i)}\right\}_{i=1}^{N}$: Set of N randomly generated numbers .

$\mathbb{D} = \{\mathbb{X}, \mathbb{Y}\}$: Dataset containing all instances and all true labels.

$\mathbb{D}_\alpha = \{\mathbb{X}, \mathbb{Z}_\alpha\}$: Dataset containing the labels given by the annotator α.

$\mathbb{D}_\alpha^{\text{train}}$, $\mathbb{D}_\alpha^{\text{test}}$: Train and test crowd datasets randomly selected from $\mathbb{D}_\alpha$ where $\mathbb{D}_\alpha^{\text{train}} U \mathbb{D}_\alpha^{\text{test}} = \mathbb{D}_\alpha$ and $\mathbb{D}_\alpha^{\text{train}} \bigcap \mathbb{D}_\alpha^{\text{test}} = \varnothing$

$f_\alpha^{(g)}(\cdot)$: Classifier g trained on dataset $\mathbb{D}_\alpha^{\text{train}}$ with random seed number g (which is also the classifier index)

$P_\alpha^{(i),(g)} = \left\{p_\alpha^{(i,k),(g)}\right\}_{k=1}^{K}$: Predicted probability set obtained in the output of the classifier $f_\alpha^{(g)}(\cdot)$ representing the probability that each class k is present in the sample.

$\theta_\alpha^{(k),(g)}$: Binarization threshold. To obtain this, we can utilize any existing thresholding technique. For example, in one technique, we analyze the ROC curve and find the corresponding threshold where the difference between the true positive rate (sensitivity) and false positive rate (1-specificity) is maximum. Alternatively, we could simply use 0.5.

$t_\alpha^{(i,k),(g)} = \begin{cases} 1 & \text{if } p_\alpha^{(i,k),(g)} \geq \theta_\alpha^{(k),(g)} \\ 0 & \text{otherwise.} \end{cases}$: Predicted label obtained by binarizing $p_\alpha^{(i,k),(g)}$.

$\eta_\alpha^{(i,k)} = \underset{g}{\text{MV}}\left(t_\alpha^{(i,k),(g)}\right)$: The output of the majority vote applied to the predicted labels obtained by the G classifiers.

$\Delta_\alpha^{(i,k)}$: Uncertainty score.

$c_\alpha^{(i,k)}$: Consistency score.

$\omega_\alpha^{(k)}$: Estimated weight for annotator α and class k.

$\nu^{(i,k)} = \frac{1}{M}\sum_\alpha \omega_\alpha^{(k)}\ \eta_\alpha^{(i,k)}$: Final aggregated label for class k and instance i.

1.3.2 Risk Calculation

Label aggregation is frequently used in various machine learning tasks, such as classification and regression, when multiple annotators assign labels to the same data points. The aggregation model refers to the underlying function that maps a set of multiple labels, obtained by different annotators, into one aggregated label. In the context of label aggregation, this model can be a neural network, a decision tree, or any other machine learning algorithm capable of learning to aggregate labels provided by multiple annotators. The objective of this study is to develop an aggregation model capable of accurately determining true labels despite potential disagreements among annotators. One common method to achieve this involves minimizing the total error (or disagreement) between the annotators' assigned labels and the true labels, as follows:

$$E = \sum_{i=1}^{N}\sum_{a=1}^{M}\left(\sum_{k=1}^{K}\delta\left(y^{(i,k)}, z_\alpha^{(i,k)}\right)\right) \tag{1.3.1}$$

where δ is the Kronecker delta function. Although error is a crucial aspect in determining the aggregation model's performance, it treats false positives and false negatives with equal weight. However, in many practical scenarios, it is essential to weigh false positives and false negatives differently depending on the specific context and potential consequences of each type of misclassification. The concept of risk allows us to achieve this by incorporating a loss function, which assigns different weights to different types of errors. In this way, risk serves as a weighted calculation of error, enabling us to better evaluate the performance of an aggregation model and its generalization capability. Let us denote loss function, $\mathcal{L}(\cdot)$, as a function that quantifies the discrepancy between the predicted labels and the true labels, accounting for the varying importance of different types of errors. Risk, denoted as $R(h)$, represents the expected value of a loss function over all possible data instances. In practice, our goal is to minimize the risk to achieve optimal performance on unseen data. However, since we only have access to a limited dataset (empirical distribution), we instead work with the empirical risk. This limitation may arise because of the need to reserve a portion of our data for testing and validation or because no dataset can fully capture all possible data instances in the real world. However, minimizing risk alone could result in overfitting, in which the aggregation model learns the noise in the training data rather than the underlying patterns, resulting in poor generalization to unseen data. To improve

generalizability, it is necessary to employ regularization techniques to strike a balance between the complexity of the aggregation model and its ability to fit the training data. Risk measurement enables us to assess the aggregation model's performance in terms of accuracy (of the aggregated labels with respect to the ground truth labels), overfitting (when risk is minimized but the model performs poorly on unseen data), and model complexity. Assume that the aggregation model $h(\cdot)$ is a function that takes a set of M label sets $Z^{(i)}$ for each instance i in the training data and calculates an aggregated label set $\widehat{Y}^{(i)}$ as an estimate of the true label set $Y^{(i)}$. Our goal is to find an aggregation model $h(\cdot)$ that minimizes risk defined as follows:

$$R(h) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(Y^{(i)}, h\left(\left\{Z_\alpha^{(i)}\right\}_{\alpha=1}^{M}\right)\right) \tag{1.3.2}$$

In this context, $\mathcal{L}(\cdot)$ represents an arbitrary loss function, which quantifies the discrepancy between predicted labels and true labels while accounting for the varying importance of different types of errors. Our goal is to choose an aggregation model $\widehat{h}$ that minimizes the risk, following the principle of risk minimization [44]:

$$\widehat{h} = \underset{h}{\operatorname{argmin}} R(h) \tag{1.3.3}$$

1.3.3 Generating Annotators' Label Sets from Ground Truth

In order to evaluate the proposed crowd-certain technique (with and without penalization) as well as other aggregation techniques, we create M fictitious annotators. To synthesize a multi-annotator dataset from a dataset with existing ground truth, we use a uniform distribution in the interval from 0.4 to 1, i.e., $\pi_\alpha^{(k)} \sim U(0.4, 1)$ (however other ranges can also be used) to obtain $M \times K$ probability thresholds Π, where K is the number of classes. (Note that an annotator may be skilled at labeling dogs, but not rabbits.) Then we use these probability thresholds to generate the crowd label set $Z_\alpha^{(i)}$ from the ground truth labels for each instance i. For each annotator α, each instance i and class k in the dataset is assigned its true label with probability $\pi_\alpha^{(k)}$ and the opposite label with probability $(1-\pi_\alpha^{(k)})$. To generate the labels for each annotator α, a random number $0 < \rho^{(i)} < 1$ is generated for each instance i in the dataset. Then $\forall \alpha, k$ if $\rho^{(i)} \leq \pi_\alpha^{(k)}$. Then the true label is used for that instance and class for the annotator α; otherwise, the incorrect label is used. The calculated annotator labels $z_\alpha^{(i,k)}$ for each annotator α, instance i and class k are as follows:

$$z_\alpha^{(i,k)} = \begin{cases} y^{(i,k)} & \text{if } \rho^{(i)} \leq \pi_\alpha^{(k)}, \\ 1 - y^{(i,k)} & \text{if } \rho^{(i)} > \pi_\alpha^{(k)}, \end{cases} \quad \forall i, a, k \tag{1.3.4}$$

To evaluate the proposed techniques over all data instances, a k-fold cross-validation is employed.

1.3.4 Uncertainty Measurement

A common approach to measure uncertainty is to increase the number of data instances X in the test dataset $\mathbb{D}_{\alpha}^{\text{test}}$ to create multiple variations of each sample data $X^{(i)}$ [45]. In this approach, for each instance i, we apply randomly generated spatial transformations and additive noise to the input data $X^{(i)}$ to obtain a transformed sample and repeat this process G times to obtain a set of G transformed samples. However, this approach is mostly suitable for cases where the input data comprises images or volume slices. Since the datasets used in this study consist of feature vectors instead of images or volume slices, this approach cannot be used. To address this problem, we introduced a modified uncertainty measurement approach, in which instead of augmenting the data instances $X^{(i)}$, we feed the same sample data to different classifiers. For the choice of classifier, we can either use a probability-based classifier such as random forest and train it under G different random states or train various classifiers and address the problem in a manner similar to ensemble learning [46] (using a set of G different classification techniques such as random forest, SVM, CNN, Adaboost, etc.). In either case, we obtain a set of G classifiers $\left\{f_{\alpha}^{(g)}(\cdot)\right\}_{g=1}^{G}$ for each annotator α. The classifier $f_{\alpha}^{(g)}(\cdot)$ is a pre-trained or pre-designed model that has been trained on a labeled training dataset $\mathbb{D}_{\alpha}^{\text{train}}$. This training process enables $f_{\alpha}^{(g)}(\cdot)$ to learn the underlying patterns in the data and make predictions on unseen instances. After training, we feed the test samples $X^{(i)} \in \mathbb{X}^{\text{test}}$ to the g-th classifier $f_{\alpha}^{(g)}(\cdot)$ as test cases. The classifier $f_{\alpha}^{(g)}(\cdot)$ then outputs a set of predicted probabilities $\left\{p_{\alpha}^{(i,k),(g)}\right\}_{k=1}^{K}$ representing the probability that class k is present in the sample. Consequently, we obtain a collection of G predicted probability sets $\left\{\left\{p_{\alpha}^{(i,k),(g)}\right\}_{k=1}^{K}\right\}_{g=1}^{G}$ for each annotator α and instance i. The set $\left\{p_{\alpha}^{(i,k),(g)}\right\}_{g=1}^{G}$ contains the predicted probabilities for class k, annotator α, and instance i. Disagreements between predicted probabilities $\left\{p_{\alpha}^{(i,k),(g)}\right\}_{g=1}^{G}$ can be used to estimate uncertainty. The reason for using classifiers rather than using the crowdsourced labels directly is two-fold. Using a probabilistic classifier helps us calculate uncertainty based on each annotator's labeling patterns that the classifier learns. Furthermore, this approach provides us with a set of pre-trained classifiers $\left\{\left\{f_{\alpha}^{(g)}(\cdot)\right\}_{g=1}^{G}\right\}_{a=1}^{M}$ that can be readily utilized on any new data instances without the need for those samples to be labeled by the original annotators. The index value $g \in \{1, 2, \ldots, G\}$ is used as the random seed value during training of the gth classifier for all annotators. Define $t_{\alpha}^{(i,k),(g)}$ as the predicted label obtained by binarizing the predicted probabilities $p_{\alpha}^{(i,k),(g)}$ using the threshold $\theta_{\alpha}^{(k),(g)}$ as shown in the Glossary of Symbols section. Uncertainty measures are used to quantify the level of uncertainty or confidence associated with the predictions of a model. In this work, we need to measure the uncertainty $u_{\alpha}^{(i,k)}$ associated with the model predictions. Some

common uncertainty measurement measures are as follows.

Entropy

Entropy is a widely used measure of uncertainty in classification problems. In an ensemble of classifiers, entropy serves as a quantitative measure of the uncertainty or disorder present in the probability distribution of the predicted class labels. A higher entropy value indicates a greater degree of uncertainty in the predictions, as the predictions of the individual classifiers in the ensemble are significantly different. In contrast, a lower entropy value indicates reduced uncertainty as the ensemble assigns very similar probabilities to a particular class, indicating strong agreement among the classifiers and increased confidence in their collective prediction. The formula for calculating entropy is as follows:

$$\Delta_{\alpha}^{(i,k)} = H\left(\left\{p_{\alpha}^{(i,k),(g)}\right\}_{g=1}^{G}\right) = -\sum_{g} p_{\alpha}^{(i,k),(g)} \log\left(p_{\alpha}^{(i,k),(g)}\right) \tag{1.3.5}$$

Standard Deviation

In regression problems, standard deviation is often used to quantify uncertainty. It measures the dispersion of predicted values around the mean. A greater standard deviation indicates greater uncertainty of the prediction. For a set of predicted values $\{t_{\alpha}^{(i,k),(g)}\}_{g=1}^{G}$ with mean value μ, the standard deviation is defined as.

$$\Delta_{\alpha}^{(i,k)} = \mathrm{SD}\left(\left\{t_{\alpha}^{(i,k),(g)}\right\}_{g=1}^{G}\right) = \sqrt{\frac{1}{G-1}\sum_{g=1}^{G}\left(t_{\alpha}^{(i,k),(g)} - \mu\right)^2}, \quad \mu = \frac{1}{G}\sum_{g=1}^{G} t_{\alpha}^{(i,k),(g)} \tag{1.3.6}$$

Predictive Interval A predictive interval provides a range within which a future observation is likely to fall with a certain level of confidence. For example, a 95% predictive interval indicates that there is a 95% likelihood that the true value falls within that range. A greater uncertainty corresponds to wider intervals. In the context of multiple classifiers, the predictive intervals can be calculated by considering the quantiles of the classifier output. For a predefined confidence level γ (e.g., 95%), for a specific class k, we need to find the quantiles Q_L^k and Q_U^k of the probability distribution of class k predicted by the G classifiers. The uncertainty can be represented by the width of the predictive interval:

$$\begin{aligned} P\left(Q_L^k \le p_{\alpha}^{(i,k),(g)} \le Q_U^k\right) &= \gamma \\ \Delta_{\alpha}^{(i,k)} &= Q_L^k - Q_U^k \end{aligned} \tag{1.3.7}$$

The steps to calculate the predictive interval are as follows:

1. Collect the class k probabilities predicted by all G classifiers for a given instance. Then sort the values in ascending order. Let us call this set $P_{\alpha}^{(i,k)} = \text{sorted}\left(\left\{p_{\alpha}^{(i,k),(g)}\right\}_{g=1}^{G}\right), \quad \forall \alpha, k, i.$

2. Calculate the lower and upper quantile indices based on the chosen confidence level γ. The lower quantile index is $L = \text{ceil}\left(\frac{G}{2}(1-\gamma)\right)$, and the upper quantile index is $U = \text{floor}\left(\frac{G}{2}(1+\gamma)\right)$, where ceil and floor are the ceiling and floor functions, respectively.

3. Find the values corresponding to the lower and upper quantile indices in the sorted $P_{\alpha}^{(i,k)}$. These values are the lower and upper quantiles Q_L^k and Q_U^k.

4. Now we have the predictive interval $P\left(Q_L^k <= p_{\alpha}^{(i,k),(g)} <= Q_U^k\right) = \gamma$, where Q_L^k and Q_U^k represent the bounds of the interval containing the α proportion of the probability mass.

Monte Carlo Dropout

The Monte Carlo dropout [47] can be used to estimate uncertainty in neural networks by applying the dropout at test time. Multiple forward passes with dropout generate a distribution of predictions from which uncertainty can be derived using any of the aforementioned techniques (standard deviation, entropy, etc.).

Bayesian Approaches

Bayesian methods offer a probabilistic framework to estimate the parameters of the model and make predictions. These methods explicitly model uncertainty by considering prior beliefs about the model parameters and then updating those beliefs based on the observed data. In Bayesian modeling, the model parameters are treated as random variables and a posterior distribution is estimated using these parameters. The following are two common Bayesian approaches for measuring the uncertainty in classification problems.

- **Bayesian model averaging (BMA):** BMA accounts for model uncertainty by combining the predictions of various models using their posterior probabilities as weighting factors. Instead of selecting a single "best" model, BMA acknowledges the possibility of multiple plausible models, each with its own strengths and weaknesses [48]. The steps to implement BMA are as follows. Select a set of candidate models that represent different hypotheses regarding the data-generating process underlying the data. These models may be of various types, such as linear regression,

decision trees, neural networks, or any other model suited to the specific problem at hand. Using the available data, train each candidate model. Calculate the posterior probabilities of the models. Using the posterior probabilities of each model as weights, calculate the weighted average of each model's predictions. The weighted average is the BMA prediction for the input instance and class.

- **Bayesian neural networks (BNNs):** BNNs [49] are an extension of conventional neural networks in which the weights and biases of the network are treated as random variables. The primary distinction between BNNs and conventional neural networks is that BNNs model uncertainty directly in the weights and biases. The posterior distributions of the network weights and biases (learned during training) capture the uncertainty, which can then be utilized to generate predictive distributions for each class. This enables multiple predictions to be generated by sampling these predictive distributions, which can be used to quantify the uncertainty associated with each class.

Committee-Based Methods

The committee-based method [50] involves training multiple models (a committee) and aggregating their predictions. The disagreement between committee members' predictions can be used as a measure of uncertainty. Examples include bagging and boosting ensemble methods and models, such as random forests.

$$\Delta_{\alpha}^{(i,k)} = \mathrm{VarCommittee}\left(P_{\alpha}^{(i,k)}\right) = \frac{1}{G-1}\sum_{g=1}^{G}\left(p_{\alpha}^{(i,k),(g)} - \mu\right)^2, \quad \mu = \frac{1}{G}\sum_{g=1}^{G} p_{\alpha}^{(i,k),(g)} \tag{1.3.8}$$

Conformal Prediction

Conformal prediction [51] is a method of constructing prediction regions that maintain a predefined level of confidence. These regions can be used to quantify the uncertainty associated with the prediction of a model. Steps to calculate the nonconformity score:

1. For each classifier g and each class k, calculate the nonconformity score. Here, $\mathrm{score\_function}$ measures the conformity of the prediction with the true label. In the context of this study, the true label can be replaced by $\eta_{\alpha}^{(i,k)}$. A common choice for $\mathrm{score\_function}$ is the absolute difference between the predicted probability and the true label, but other options can be used depending on the specific problem and requirements. Define the nonconformity score as $\zeta_k^g = \mathrm{score\_function}\left(p_{\alpha}^{(i,k),(g)}, y^{(i,k)}\right)$

2. Calculate the p-value for each class k as the proportion of classifiers with nonconformity scores greater than or equal to a predefined threshold $\mathsf{T}^{(k)}$: p-values$(k) = \frac{\left|\{g:\, \zeta^{(k),(g)} \geq \mathsf{T}^{(k)}\}\right|}{G}$

3. The p-values calculated for each class k represent the uncertainty associated with that class. A higher p-value indicates a higher level of agreement among the classifiers for a given class, whereas a lower p-value suggests greater uncertainty or disagreement.

The uncertainty measures discussed above are only some of the available options. Selecting an appropriate measure depends on factors such as the problem domain, the chosen model, and the specific requirements of a given application. For this study, we use the variance technique shown in Equation (1.3.6) as our uncertainty measurement due to its simplicity. However, other measures could also be employed as suitable alternatives.

1.3.5 Crowd-Certain: Uncertainty-Based Weighted Soft Majority Voting

Consistency Measurement

Define $c_{\alpha}^{(i,k)}$ as the consistency score for annotator α, class k and instance i. We calculate this consistency score using the uncertainty score $\Delta_{\alpha}^{(i,k)}$ explained in the previous section. We use two approaches to calculate $c_{\alpha}^{(i,k)}$ from $\Delta_{\alpha}^{(i,k)}$.

1. The first approach is to simply subtract the uncertainty from 1 as follows:

$$c_{\alpha}^{(i,k)} = 1 - \Delta_{\alpha}^{(i,k)} \quad , \; \forall i, \alpha, k \tag{1.3.9}$$

2. In a second approach (shown in Equation (1.3.10)), we penalize annotators for instances in which their predicted label $\eta_{\alpha}^{(i,k)}$ (explained in the Glossary of Symbols section) does not match the MV of all annotator labels $\underset{\alpha}{\mathrm{MV}}\left(z_{\alpha,\,k}^{(i,k)}\right)$. As previously discussed, instead of directly working with the annotator's labels $z_{\alpha}^{(i,k)}$, we use the predicted labels obtained from the ensemble of classifiers $\eta_{\alpha}^{(i,k)}$. This methodology does not require repeating the crowd-labeling process for new data samples. In particular, we are likely not to have access to the same crowd of annotators employed in the training dataset.

$$c_{\alpha}^{(i,k)} = \begin{cases} 1 - \Delta_{\alpha}^{(i,k)} & \text{if } \eta_{\alpha}^{(i,k)} = \mathrm{MV}_{\alpha}(\eta_{\alpha}^{(i,k)}) \\ 0 & \text{otherwise} \end{cases} \tag{1.3.10}$$

Reliability Measurement

For each annotator, for each class, and for each instance, there is a consistency score, $c_\alpha^{(i,k)}$. By averaging these scores across all instances, we can define a reliability score for each annotator and for each class:

$$\psi_\alpha^{(k)} = \frac{1}{N} \sum_{i=1}^{N} c_\alpha^{(i,k)} \tag{1.3.11}$$

If desired, one may also calculate an overall reliability score for each annotator by averaging across all classes:

$$\psi_\alpha = \frac{1}{K} \sum_{k=1}^{K} \psi_\alpha^{(k)} \tag{1.3.12}$$

Weight Measurement

Furthermore, we calculate the annotators' weights $\omega_\alpha^{(k)}$ for each class k by normalizing the reliability values as follows:

$$\omega_\alpha^{(k)} = \frac{\psi_\alpha^{(k)}}{\sum_{\alpha=1}^{M} \psi_\alpha^{(k)}} \tag{1.3.13}$$

Aggregated Label Calculation

Finally, the aggregated label $\nu^{(i,k)}$ for each instance i and class k is the weighted average of the predicted labels $\eta_\alpha^{(i,k)}$ for each annotator α:

$$\nu^{(i,k)} = \begin{cases} 1 & \text{if } \left(\sum_{\alpha=1}^{M} \omega_\alpha^{(k)} \eta_\alpha^{(i,k)}\right) > 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, k \tag{1.3.14}$$

Confidence Score Calculation

In previous section we showed how to calculate the aggregated label $\nu^{(i,k)}$ (shown in Equation (1.3.14)). Define $F^{(i,k)}$ as the confidence score for instance i and class k. We calculate two confidence scores $F^{(i,k)}$, based on how many different annotators agree on the reported label $\nu^{(i,k)}$. The confidence scores show the level of confidence we should place on the aggregated labels. To calculate this confidence score, we modify the two techniques used by Sheng [42] and Tao [1] to incorporate our calculated weight $\omega_\alpha^{(k)}$ shown in Equation (1.3.13) for each worker α.

Using Weighted Sum: In a standard voting system, for every instance i and class k, each contributor in the

group—whether that be an annotator in a crowd or a model in an ensemble learning context—provides a class label. The label receiving the most votes, meaning it's 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 factor 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 weight to votes with greater confidence, or, in other words, votes with greater weight. This refined approach prioritizes confidence, thereby enhancing the ensemble's overall effectiveness.

The confidence score $F_{\Omega}^{(i,k)}$ is formulated as follow.

$$F_{\Omega}^{(i,k)} = \sum_{\alpha=1}^{M} \omega_{\alpha}^{(k)} \delta \left(\eta_{\alpha}^{(i,k)} \ , \ \ \nu^{(i,k)} \right) \tag{1.3.15}$$

where δ is the Kronecker delta function.

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 you are testing a hypothesis concerning a parameter of a population, and you collect a sample of observations. 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, A and B. 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).

In the following equation, the probability of obtaining k successes or more out of n trials where the probability of success on any given trial is p is calculated. In this context, "success" could refer to the event in question, such as the correct classification of an object or the acceptance of a hypothesis. The CDF of the beta distribution at the decision threshold of 0.5 is used to calculate a confidence score $F_{\beta}^{(i,k)}$. To calculate the two shape parameters of the beta distributions $l^{(i,k)}$ and $u^{(i,k)}$, a weighted sum of all correct and incorrect aggregated labels, is used respectively:

$$\begin{aligned} l^{(i,k)} &= 1 + \sum_{\alpha=1}^{M} \omega_{\alpha}^{(k)} \; \delta\left(\eta_{\alpha}^{(i,k)}, \nu_{k}^{(i,k)}\right) \\ u^{(i,k)} &= 1 + \sum_{\alpha=1}^{M} \omega_{\alpha}^{(k)} \; \delta\left(\eta_{\alpha}^{(i,k)}, 1 - \nu_{k}^{(i,k)}\right) \end{aligned} \tag{1.3.16}$$

$$F_{\beta}^{(i,k)} = I_{0.5}\left(l^{(i,k)}, u^{(i,k)}\right) = \sum_{t=[l^{(i,k)}]}^{T-1} \frac{(T-1)!}{t!(T-1-t)!} 0.5^{T-1} \tag{1.3.17}$$

where $T = \left[l^{(i,k)} + u^{(i,k)}\right)]$ and $[\cdot]$ is an integer function.

1.3.6 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:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \delta\left(\nu^{(i,k)}, y^{(i,k)}\right) \tag{1.3.18}$$

  where δ is the Kronecker delta function, N is the total number of instances, and K is the number of classes, $y^{(i,k)}$ and $\nu^{(i,k)}$ are the ground truth and aggregated label respectively for class k and instance i. Although accuracy is most effective for balanced classes, 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's computed as:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1.3.19}$$

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

- **Area Under the Receiver Operating Characteristic Curve (AUC-ROC):** AUC-ROC measures the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity) for every possible cut-off. 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's calculated as the mean squared difference between the predicted probability and the actual outcome, thereby rewarding predictions that are both well-calibrated and confident. It can be calculated as follows:

$$\text{Brier Score} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \left(v^{(i,k)} - y^{(i,k)} \right)^2 \tag{1.3.20}$$

- **Expected Calibration Error (ECE):** ECE is used to quantify the calibration of the confidence scores produced by a model. It's 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:

$$\text{ECE} = \sum_{b=1}^{B} \frac{|B_b|}{N} \left| \text{Accuracy}(B_b) - \text{Confidence-Score}(B_b) \right| \tag{1.3.21}$$

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.

1.4 Results

To evaluate our proposed technique, we conducted a series of experiments comparing the proposed technique with several existing techniques such as MV, Tao [1], and Sheng [42], as well as with other

crowdsourcing methodologies reported in the crowd-kit package [52] including Gold Majority Voting, MMSR [53], Wawa, Zero-Based Skill, GLAD [54], and Dawid Skene [15].

1.4.1 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 1.1 provides an overview of the datasets used. All datasets are obtained from the University of California, Irvine (UCI) repository [55].

Table 1.1: Descriptions 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 annotator 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 annotator 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.

1.4.2 Benchmarks

Tao [1] and Sheng [42] techniques were implemented in Python to evaluate their performance. Furthermore, the crowd-kit package (A General-Purpose Crowdsourcing Computational Quality Control Toolkit for Python) [52] was used to implement the remaining benchmark techniques, including Gold Majority Voting, MMSR [53], Wawa, Zero-Based Skill, GLAD [54], and Dawid Skene [15].

- **Worker Agreement with Aggregate (WAWA) [56]:** (Annotator Agreement with Aggregate), also referred to as "inter-rater agreement", is a commonly used statistic for non-testing problems. This indicates the average frequency with which each annotator's response matches the aggregate response for each instance.

- **Zero-Based-Skill (ZBS)** employs a weighted majority vote (WMV). After processing a collection of instances, it re-evaluates the abilities of the annotators 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):** Iterative algorithm that calculates the log-likelihood of the task being positive while modeling the reliabilities of the workers. Let A_{ij} be a matrix of answers of worker j on task i. $A_{ij} = 0$ if worker j didn't answer the task i, otherwise $|A_{ij}| = 1$. The algorithm operates on real-valued task messages $x_{i\to j}$ and worker messages $y_{j\to i}$. A task message $x_{i\to j}$ represents the log-likelihood of task i being a positive task, and a worker message $y_{j\to i}$ represents how reliable worker j is. On iteration k the values are updated as follows:

$$x_{i\to j}^{(k)} = \sum_{j'\in\partial i\setminus j} A_{ij'} y_{j'\to i}^{(k-1)} \qquad y_{j\to i}^{(k)} = \sum_{i'\in\partial j\setminus i} A_{i'j} x_{i'\to j}^{(k-1)} \tag{1.4.1}$$

- **Multi-Annotator Competence Estimation (MACE) [57]:** 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 yield a correct label. If the worker is spamming, they answer according to their probability distribution. Let's assume that the correct label T_i comes from a discrete uniform distribution. When a worker annotates the task, they are in the spamming state with probability $\mathrm{Bernoulli}(1-\theta_w)$. So, if their state $s_w = 0$, their

response $A_{iw} = T_i$. Otherwise, their response A_{iw} is drawn from a multinomial distribution with parameters ξ_w.

- **Matrix Mean-Subsequence-Reduced Algorithm (MMSR) [53]:** The M-MSR assumes that workers have different level of expertise and associated with a vector of "skills" $\boldsymbol{s}$ which entries s_i show the probability of the worker i to answer correctly to the given task. Having that, we can show that.

$$\mathbb{E}\left[\frac{M}{M-1}\widetilde{C} - \frac{1}{M-1}\mathbf{1}\mathbf{1}^T\right] = \boldsymbol{s}\boldsymbol{s}^T, \tag{1.4.2}$$

where M 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 M-MSR algorithm is an iterative algorithm for *rubust* 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 $\log \frac{(M-1)s_i}{1-s_i}$.

- **Generative model of Labels, Abilities, and Difficulties (GLAD) [54]:** A probabilistic model that parametrizes workers' abilities and tasks' dificulties. Let's consider a case of K class classification. Let p be a vector of prior class probabilities, $\alpha_i \in (-\infty, +\infty)$ be a worker's ability parameter, $\beta_j \in (0, +\infty)$ be an inverse task's difficulty, z_j be a latent variable representing the true task's label, and y_j^i be a worker's response that we observe. The relationships between this variables and parameters according to GLAD are represented by the following latent label model. The prior probability of z_j being equal to c is $\Pr(z_j = c) = p[c]$, 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 worker's ability and inverse task's difficulty:

$$\Pr(y_j^i = k | z_j = c) = \begin{cases} a(i,j), & k = c \\ \frac{1-a(i,j)}{K-1}, & k \neq c \end{cases}, \tag{1.4.3}$$

where $a(i,j) = \frac{1}{1+\exp(-\alpha_i\beta_j)}$.

Parameters p, α, β and latent variables z are optimized through the Expectation-Minimization algorithm.

- **Dawid-Skene [15]:** Probabilistic model that parametrizes workers' level of expertise through confusion matrices. Let e^w be a worker's confusion (error) matrix of size $K \times K$ in case of K class classification, p be a vector of prior classes probabilities, z_j be a true task's label, and y_j^w be a worker's answer for the task j. The relationships between these parameters are represented by the following latent label model. Here the prior true label probability is $\Pr(z_j = c) = p[c]$ and the distribution on the worker's responses given the true label c is represented by the corresponding column of the error matrix: $\Pr(y_j^w = k | z_j = c) = e^w[k, c]$ Parameters p and e^w and latent variables z are optimized through the Expectation-Maximization algorithm.

- Descriptions of the other techniques can be found in their respective references.

1.4.3 Weight Measurement Evaluation

Following the generation of multi-label sets, the aggregate labels were determined using both the proposed approach and various established methods. We examined two strategies for classifier selection, as detailed in Section 1.3.4. Despite there being no substantial variations in the final outcomes, 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 annotator 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 1.3.

Figure 1.1 depicts the relationship between the randomly assigned annotators' 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 an annotator 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.

1.4.4 Label Aggregation Evaluation

The Figure 1.2 portrays a thorough accuracy comparison of our novel 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 1.3.3.

For a comprehensive evaluation, all experiments were repeated three times using different random

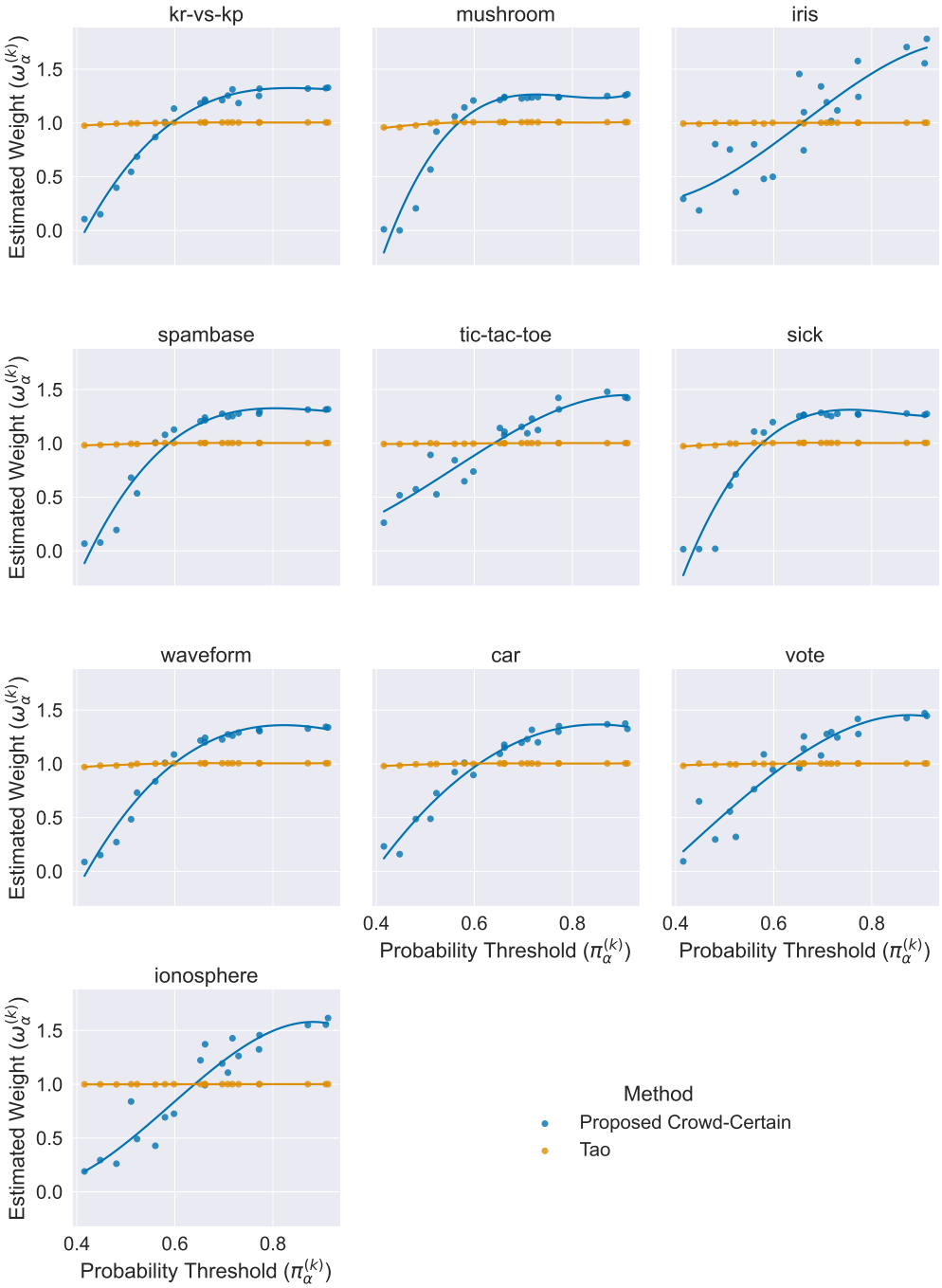

Figure 1.1: Comparison of weight computation techniques across ten different datasets. Each subplot corresponds to a unique dataset, illustrating the relationship between the randomly assigned annotator's probability threshold ($\pi_\alpha^{(k)}$) (horizontal axis) and the computed weights ($\omega_\alpha^{(k)}$) (vertical axis) for the proposed aggregation technique with penalization "crowd-certain" and Tao [1]. The individual data points represent actual measured weights, while the curve stands for the regression line.

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, "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 case of Tao [1] the aggregated labels were obtained using the equation 1.4.4. These methods inherently involve re-running simulations for every new dataset, which could be computationally expensive and time-consuming.

$$\nu^{(i,k)} = \begin{cases} 1 & \text{if } \left(\sum_{\alpha=1}^{M} \omega_{\alpha}^{(k)} \, z_{\alpha}^{(i,k)}\right) > 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, k \tag{1.4.4}$$

Across all ten datasets, it is clear that the "Crowd-Certain" method consistently outperforms the existing methods, yielding higher average accuracy rates. 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 'tic-tac-toe', where the "Crowd-Certain" method achieves superior average accuracies of around 0.980, 0.900, and 0.741, respectively. In contrast, the competing methods struggled to exceed an average accuracy of 0.771 in these datasets.

We further extended our experiment to explore the effects of varying the number of annotators, ranging from 3 up to 7. The results shown in Figure 1.3, 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 annotators. These plots provide a clear visual summary of our technique's robust performance across various settings, including the median, quartiles, and potential outliers in the distribution of accuracies. Notably, our proposed "Crowd-Certain" technique consistently shows improvements over the 10 benchmark methods across all scenarios. This enhancement is evident irrespective of the number of annotators involved, further highlighting the robustness and adaptability of the "Crowd-Certain" approach.

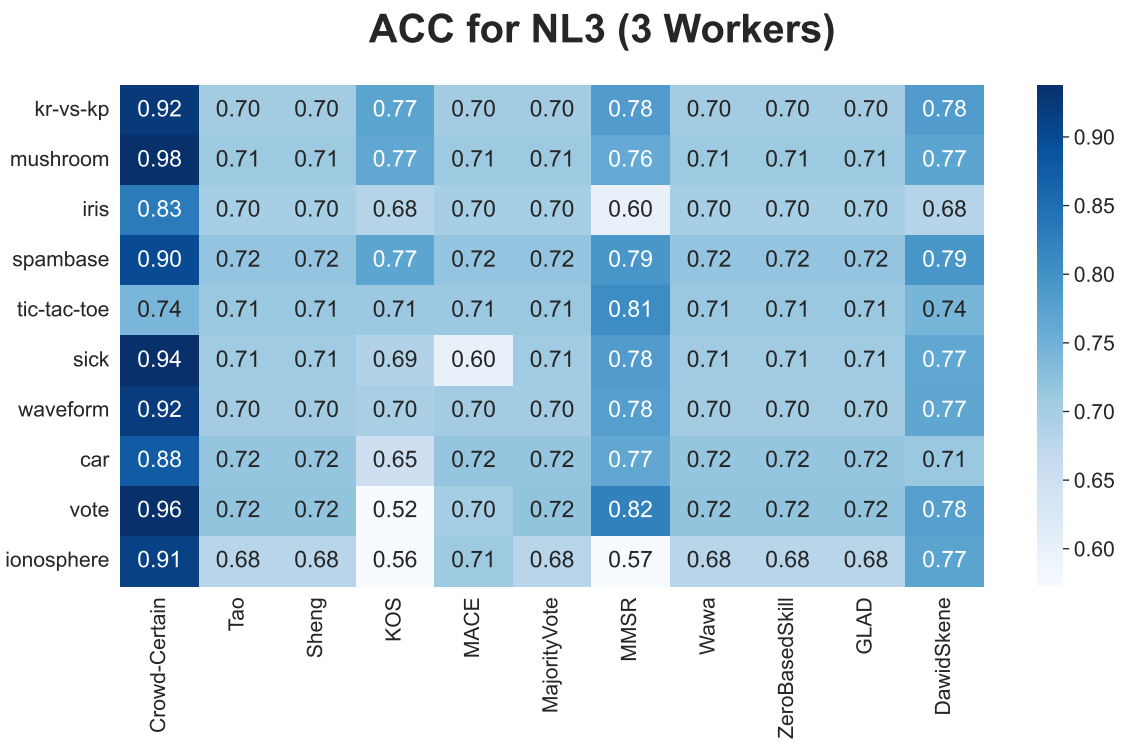

Figure 1.2: Comparison of Accuracy Scores for Multiple Label Aggregation Techniques on Various Datasets. The figure displays the mean accuracy score obtained across three independent trials for the proposed method ("Crowd-Certain") and ten existing label aggregation techniques. The trials were conducted using three labelers (workers) per dataset. The aggregated labels for "Crowd-Certain" were derived using predicted probabilities, allowing for reuse of trained classifiers. In contrast, existing techniques used actual crowd labels, necessitating repeated simulations.

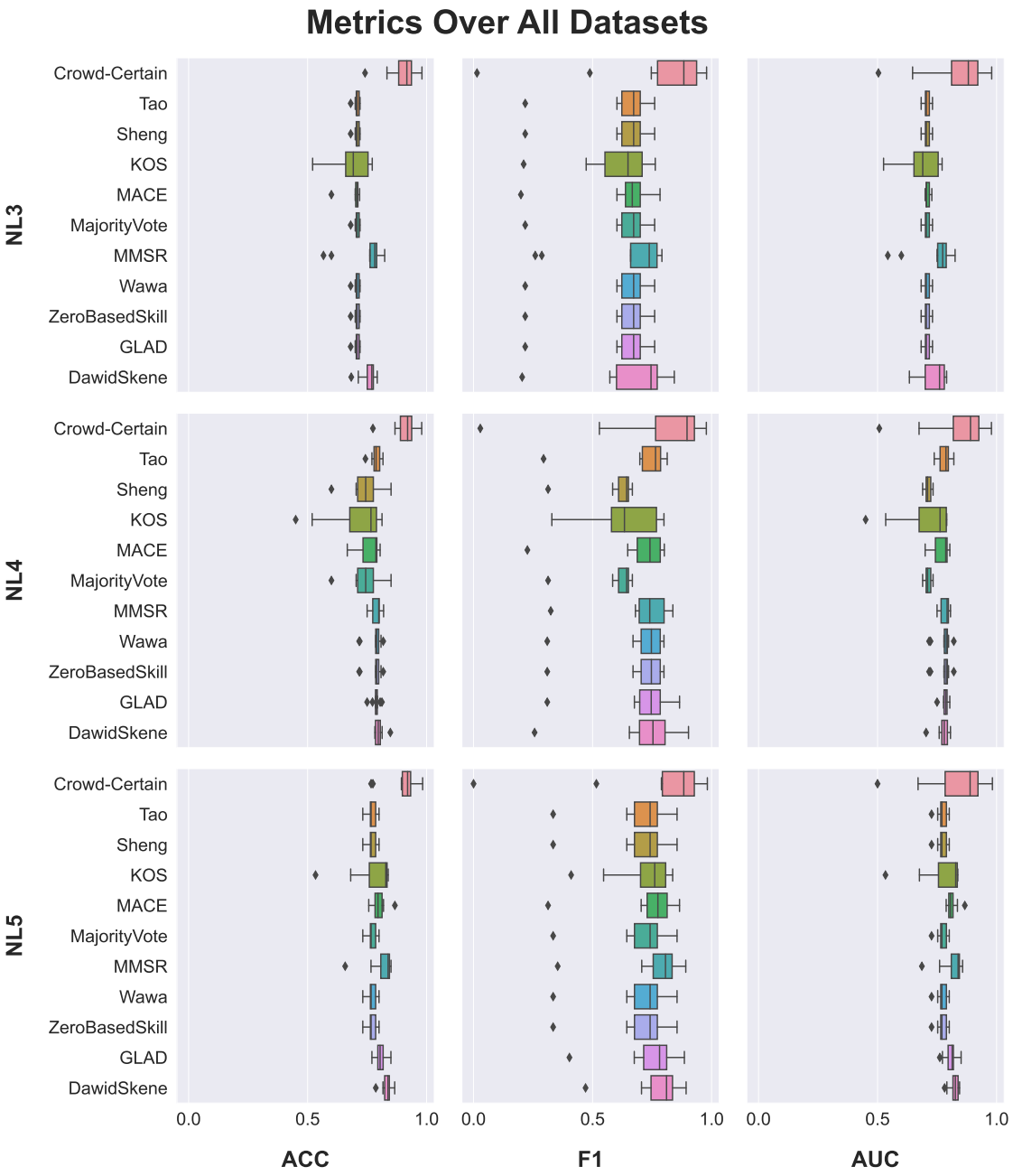
Metrics Over All Datasets
NL3
NL4
NL5
Crowd-Certain
Tao
Sheng
KOS
MACE
MajorityVote
MMSR
Wawa
ZeroBasedSkill
GLAD
DawidSkene
0.0
0.5
1.0
ACC
F1
AUC

Figure 1.3

1.4.5 Confidence Score Evaluation

The table 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 Loss. The evaluations were conducted across a variety of datasets and three techniques: Crowd-Certain, Tao, and Sheng, when using 3 workers.

Figure 1.4 depicts the performance of three different strategies: Crowd-Certain, Tao, and Sheng, compared across two metrics - Expected Calibration Error (ECE) and Brier Score Loss. These results are obtained using two different confidence score calculation techniques “Freq” and “Beta”, applied over ten different datasets when having three annotators. The ECE score offers an aggregated measure of the reliability of probabilistic predictions. In this case, it is used to assess the calibration of the aggregated labels across different techniques and strategies. A lower ECE score indicates better-calibrated predictions, i.e., the predicted probabilities are closer to the true probabilities. Brier Score Loss is a metric that quantifies the accuracy of probabilistic predictions. It calculates the mean squared difference between the predicted probabilities and the actual outcome. Hence, lower Brier Score Loss values correspond to better model performance. In the Figure 1.4, it can be observed that for the ECE metric, across all datasets, the proposed Crowd-Certain strategy consistently achieves lower scores when compared to Tao and Sheng, for both “Freq” and “Beta” techniques. This indicates that the Crowd-Certain strategy offers better-calibrated predictions, providing a higher level of confidence in the aggregated labels. For the Brier Score Loss metric, the Crowd-Certain strategy also appears to outperform Tao and Sheng across most datasets, for both techniques..

The Figure 1.5 showcases the results for two metrics, Expected Calibration Error (ECE) and Brier Score Loss, for two confidence measurement techniques - “Freq” and “Beta” 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 labelers from 3 (denoted with NL3) up to (denoted with NL7).

In general, the ECE and Brier Score Loss both increase as the number of labelers increases, which suggests that increasing the number of labelers 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 consistently yields lower ECE and Brier Score across different numbers of labelers compared to the Tao and Sheng techniques, indicating better calibration and sharper predictions. For the beta strategy, the performance varies between techniques. The Tao technique generally results in higher ECE and Brier Score Loss, indicating worse calibration

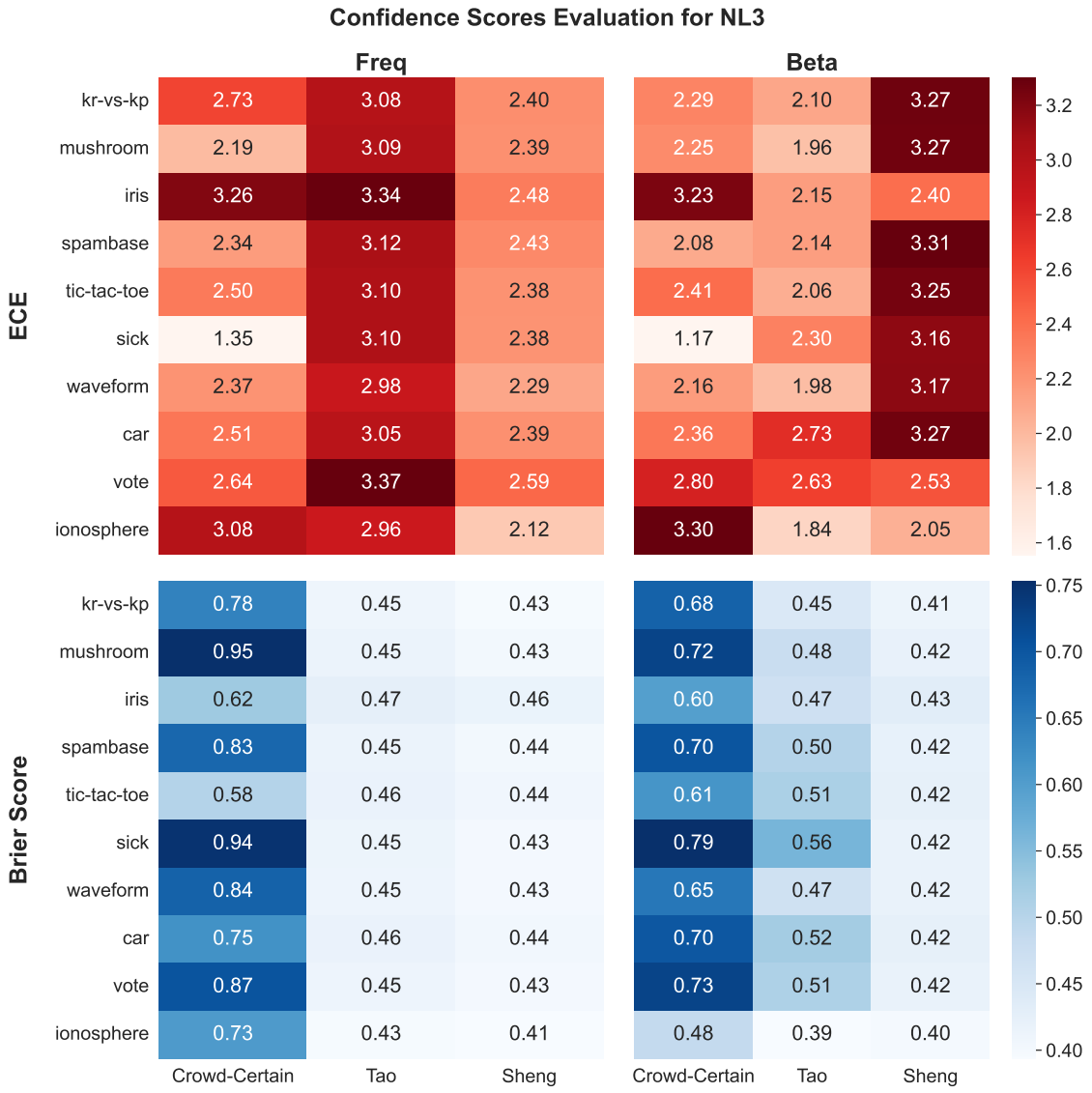

Figure 1.4: Comparison of Expected Calibration Error (ECE) and Brier Score Loss for two confidence score measurement strategies ("Freq" and "Beta") across three different techniques (Crowd-Certain, Tao, and Sheng). Results are shown for ten different datasets for 3 workers (NL3). The metrics reflect the calibration and sharpness of the predictions under different configurations.

and sharper predictions, whereas the Crowd-Certain and Sheng techniques show varying performance depending on the number of labelers.

For the Brier Score Loss, the "Freq" strategy combined with the Crowd-Certain technique tends to perform better across all numbers of labelers 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 labelers, indicating a good match between predicted confidences and observed frequencies. However, the ECE tends to increase as the number of labelers increases, indicating a decline in calibration.

Overall, these results suggest that the choice of the confidence measurement technique and the strategy has significant impacts on the calibration and sharpness of the predictions. Further investigations could be beneficial to understand the specific conditions under which certain techniques and strategies yield superior performance.

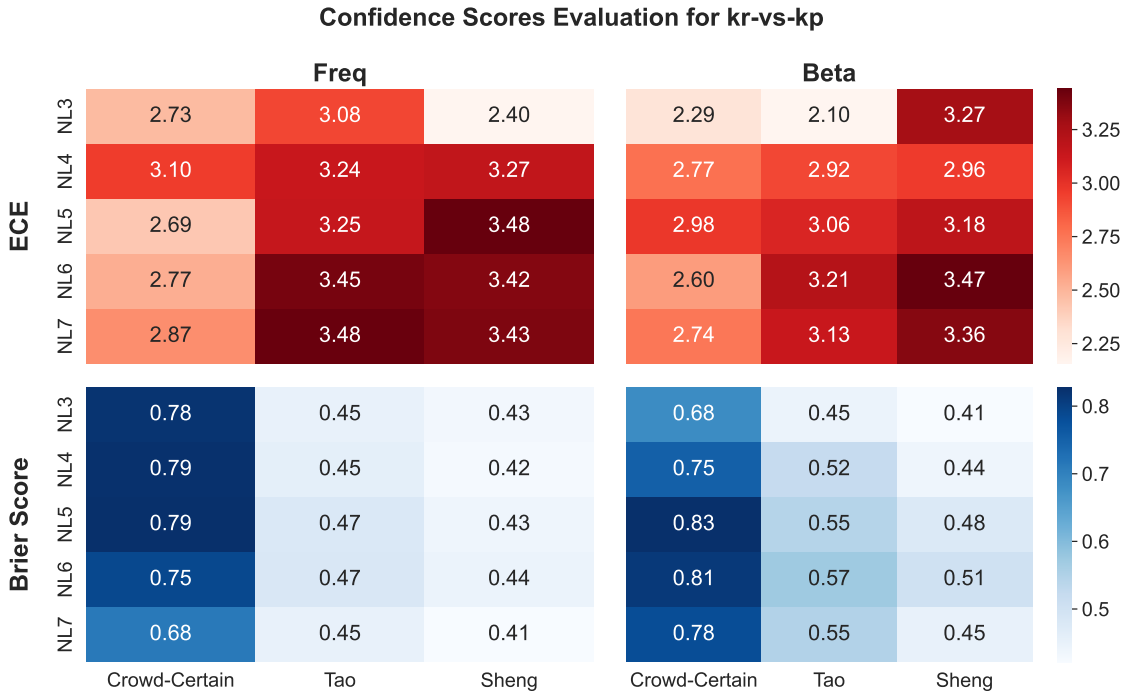

Figure 1.5: Comparison of Expected Calibration Error (ECE) and Brier Score Loss for two confidence score measurement strategies ("Freq" and "Beta") across three different techniques (Crowd-Certain, Tao, and Sheng). Results are shown for varying numbers of labelers (NL3 to NL7) on the kr-vs-kp dataset. The metrics reflect the calibration and sharpness of the predictions under different configurations.

1.5 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 annotators. In response to this, we have developed a novel label aggregation method that calculates a reliability score for each annotator based on the annotator's consistency versus a trained classifier. In the first approach (proposed techniques without penalization), we utilize uncertainty estimates to assign each annotator a more accurate weight, which correlates with their agreement with others and their consistency during labeling. In the second approach (proposed technique with penalization. Also noted as crowd-certain), we improve on this technique by penalizing the annotator for their disagreements with other annotators (shown in Eq. 1.3.10 and hence mitigate the effect of annotator's bias when calculating the final weights $\omega_{\alpha}^{(k)}$). The first part (calculating weights based on annotator's consistency) of the proposed crowd-certain algorithm (the proposed technique with penalization) is essential because non-expert annotators often exhibit more irregular consistency during labeling than experts, as they are not well trained to identify specific features. This (utilizing consistency when assigning weights to each annotator) helps to differentiate expert and non-expert annotators. The goal of the second part (penalty for voting against the majority) of the algorithm is to prevent the algorithm from assigning disproportionately high weights to annotators who are consistently incorrect. For example, if annotators consistently mislabel a specific bird species, the second part penalizes them for their error, despite their consistency. Furthermore, our method reports a single weight for the entire dataset instead of individual weights for each instance. This enables the reuse of calculated weights for future unlabeled test samples without needing to reacquire labels or retrain classifiers each time new data need labeling. While we have not assessed our method in multi-label scenarios, the proposed techniques are anticipated to perform comparably on multi-label datasets, considering that all steps of the proposed approach involve per-class calculations. Experiments conducted on various crowdsourcing datasets demonstrate that our proposed methods outperform existing techniques in terms of accuracy and variance, especially when there are few annotators available.

1.6 Availability of Data and Materials

The code can be found in crowd-certain

1.7 Appendices

List of abbreviations

Competing interests

Acknowledgements

Chapter 2

A Hierarchical Multilabel Classification Method for Enhanced Thoracic Disease Diagnosis in Chest Radiography

Accurate diagnosis of thoracic diseases from chest radiographs is a challenging task that can lead to diagnostic errors and negative patient outcomes. This study introduces two novel hierarchical multi-label classification methods, leveraging the taxonomy of pathologies to boost both the accuracy and interpretability of disease classifications. These methods cater to scenarios where ground truth is accessible (termed "loss") as well as when it isn't (termed "logit"). By utilizing disease taxonomy, the proposed methods acknowledge the interrelationships among diseases, thereby enhancing their adaptability to new tasks. The "logit" method offers ease of integration with existing pre-trained models, eliminating the need for re-optimization and ensuring extensive applicability. Conversely, the "loss" method modifies the loss function during the training phase, thus providing an avenue for integration into the current training process. The proposed techniques were evaluated on three publicly accessible, diverse chest radiograph datasets, namely CheXpert, PadChest, and NIH Chest-Xray14 as well as various statistical tests. The results underpin the significant enhancement in accuracy and interpretability these methods provide in the diagnosis of thoracic diseases in chest radiography. This approach has the potential to promote an accurate and efficient diagnosis by providing an additional layer of decision support to radiologists, ultimately leading to better patient outcomes.

KEYWORDS: Chest radiography, hierarchical classification, disease taxonomy, multilabel classification, conditional loss function, diagnostic errors, machine learning, medical imaging

2.1 Introduction

Chest radiography (CXR) is a prevalent radiological examination for diagnosing lung and heart disorders, constituting a significant share of ordered imaging studies. Fast and accurate detection of different thoracic diseases, such as pneumothorax, is crucial for optimal patient care [58]. However, interpreting CXRs can be challenging due to similarities between different thoracic diseases, which may result in misinterpretation even by experienced radiologists [59]. Consequently, devising an accurate system to identify and localize common thoracic diseases can aid radiologists in minimizing diagnostic errors [60, 61]. Progress in natural language processing (NLP) has enabled the collection of extensive annotated datasets such as ChestX-ray8 [62], PADCHEST [63], and CheXpert [64], allowing researchers to develop more efficient and robust supervised learning algorithms. Convolutional neural networks (CNNs) exhibit potential for learning intricate relationships between image objects. However, their training necessitates vast amounts of labeled data, which can be both expensive and time-consuming to acquire. Despite these challenges, deep learning techniques have become increasingly popular in medical imaging, especially in radiology, due to their ability to perform complex tasks with minimal human intervention [65].

The timely diagnosis and effective treatment of diseases depend on the fast and accurate detection of anomalies in medical images. Deep learning techniques have made substantial progress in the medical imaging domain, exhibiting impressive success across various applications [66, 67]. Although recent advances in deep learning have facilitated the creation of CAD systems capable of classifying and localizing prevalent thoracic diseases using CXR images, most of these techniques have concentrated on specific diseases [68, 69, 70, 71], leaving ample opportunities to investigate a unified deep learning framework that can efficiently detect a broad spectrum of common thoracic diseases. Further, conventional classification methods are primarily designed for single-label predictions and struggle with multi-label classification, which requires predicting multiple labels for each input sample. In multi-label classification, common methods like the One-vs-All (OVA) approach exhibit limitations, including high computational complexity and an inability to capture intricate label relationships [72].

This paper aims to tackle the challenges of multi-label classification by introducing a hierarchical framework that incorporates the relationships between different classes to provide a more accurate classification framework. Two different approaches are proposed for scenarios where ground truth is available, in which the proposed technique is applied to the loss function, and for scenarios where ground truth is not available, in which it is applied to the logit values. The latter provides a transfer learning approach that improves classification accuracy without necessitating costly computational

resources. The rest of this paper is structured as follows. Section 2 discusses related work on multi-label classification and hierarchical loss functions; Section 3 describes the proposed techniques for integrating label hierarchy into multi-label classification techniques; Section 4 presents experimental results using the chest radiograph dataset; and Section 5 concludes the paper and outlines future research directions.

2.2 Related Work

The introduction of the ChestX-ray8 dataset and its associated model [62] marked a significant advancement in large-scale CXR classification, leading to numerous improvements in both modeling and dataset collection. These enhancements include the integration of ensemble methods [73], attention mechanisms [74, 75], and localization techniques [76, 77, 78, 79]. Most early approaches use "binary relevance" (BR) learning, which reduces the multi-label classification problem to binary classification by training a binary classifier for each class [80]. However, BR-based techniques do not account for label dependence, either conditional (Instance-specific label dependence) where in a given instance, the presence or absence of one label may impact another's or marginal (dataset-specific label dependence) where certain labels may co-occur more frequently [81].

Multi-label classification, unlike multi-class methods, classifies instances into multiple categories simultaneously. For example, a single chest radiograph image can have both Edema and Cardiomegaly [72,82]. Significant research on integrating taxonomies through hierarchical classification was conducted prior to the advent of deep learning by extracting a set of binary hierarchical multi-label classification (HMLC) labels from pseudo-probability predictions [83]. Early methods used hierarchical and multi-label generalizations of traditional algorithms, such as nearest-neighbor or multi-layer perceptron [84] and decision trees [85]. With the rise of deep learning, the adaptation of convolutional neural networks (CNN) for hierarchical classification has gained increasing attention [86, 87, 88, 89].

Hierarchical multi-label Classification Technique

In many cases, the diagnosis or observation of a particular condition on a CXR (or other medical imaging data) is dependent on the presence or absence of the parent class [90]. For example, if a radiologist is trying to diagnose pneumonia in a patient, they may first look for evidence of lung consolidation (parent label) in the CXR. Consequently, it is possible to make more accurate diagnoses by taking into account the relationship between labels. However, many existing CXR classification methods do not consider the dependence between labels and instead treat each label independently. These algorithms

are known as "flat classification" methods [91]. Furthermore, some labels at the lower levels of the hierarchy, specifically leaf nodes, have very few positive examples, making the flat learning model susceptible to negative class bias. To address these issues, we must create a model that considers the hierarchical nature of the CXR.

Hierarchical multi-label classification methods have been successfully implemented in a variety of domains, including text processing [92], visual recognition [93], and genomic analysis [83]. A common technique [94] for exploiting such a hierarchy is to train a classifier on conditional data while ignoring all samples with negative parent-level labels and then reintroducing these samples to fine-tune the network across the entire dataset [94]. These approaches help the classifier focus on the relevant data during initial training, thus improving the prediction accuracy. However, these techniques are computationally expensive, as they require training a classifier on conditional data and then fine-tuning it on a full dataset. This makes them difficult to apply to real-world problems, where the amount of data is often very large. Another common strategy is cascading architecture where different classifiers are trained at each level of the hierarchy. Although these techniques enable more granular data analysis (each classifier can focus on a specific level of the hierarchy), they require a substantial amount of computational resources. Other existing deep learning-based approaches often use complex combinations of CNNs and recurrent neural networks (RNNs) [86, 87].

We propose a method that takes advantage of hierarchical relationships between labels without imposing computational requirements. Our proposed method is adaptable to the computational capacity of the user. If sufficient computational resources are available, it can be used as a standalone loss function during the optimization process, or it can be applied to test samples without the need to fine-tune the pre-trained ML model.

2.3 Methods

We propose a novel method that improves the accuracy and interpretability of multi-label classification with applications such as chest radiograph (CXR). Two different approaches are proposed. In the first approach, which requires access to ground truth labels, the hierarchical relationships between different classes are embedded into the loss function. In a second approach, the hierarchical relationships are used to update the value of logits prior to the calculation of predicted probabilities for each class. As a transfer learning approach, these two techniques facilitate the adoption and/or fine-tuning of pre-trained models, thereby augmenting their generalizability to novel tasks. This ultimately contributes to the improvement of disease diagnosis and treatment through increased accuracy within applications

where there is a hierarchical relationship between abnormalities.

One of the key benefits of the proposed techniques is the enhancement of interpretability. By organizing diseases into a hierarchical structure and leveraging their relationships, the model not only improves classification performance, but also provides insights into the relationships among predicted diseases. This additional layer of interpretability can help radiologists understand the rationale behind the model predictions, build trust in the model output, and facilitate its integration into clinical workflows. Furthermore, the hierarchical nature of the taxonomy allows radiologists to explore predictions at various levels of granularity, depending on the level of detail required for a specific case.

2.3.1 Glossary of Symbols

Let us denote the following parameters:

- $\mathcal{C} = \{c_k\}_{k=1}^{K}, c_k \in \{0, 1\}$: the set of classes (categories) in the multi-label dataset, where c_k is the name of the k-th class.
- $\mathcal{E}$: set of directed edges representing parent-child relationships between classes.
- $\mathcal{G} = \{\mathcal{C}, \mathcal{E}\}$: Directed acyclic graph (DAG) $\mathcal{G}$ representing the taxonomy of thoracic diseases.
- $c_j = \Lambda(c_k) \in \mathcal{C}$: parent class of class c_k in DAG $\mathcal{G}$.
- $\mathcal{J}(c_j) \subset \mathcal{C}$: set of child classes of class c_j in DAG $\mathcal{G}$
- $y_k^{(i)} \in \{0, 1\}$: true label for the k-th class of instance i.
- $q_k^{(i)} \in (-\infty, 0)$: logits obtained in the last layer of the neural network model before the sigmoid layer.
- $p_k^{(i)} = \text{sigmoid}\left(q_k^{(i)}\right) = \frac{1}{1+\exp\left(-q_k^{(i)}\right)}$: predicted probability for the k-th class (c_k) of instance i with a value between 0 and 1. $p_k^{(i)}$ represents the likelihood that class k is present in instance i and is obtained by passing logits $q_k^{(i)}$ through a sigmoid function.
- θ_k: Binarization threshold for class k. To obtain this, we can utilize any existing thresholding technique (for example, in one technique, we analyze the ROC curve and find the corresponding threshold where the difference between the true positive rate (sensitivity) and false positive rate

(1-specificity) is maximum; Alternatively, we could simply use 0.5).

- $t_k^{(i)} = \begin{cases} 1 & \text{if } p_k^{(i)} \geq \theta_k \\ 0 & \text{otherwise.} \end{cases}$: predicted label obtained by binarizing the $p_k^{(i)}$
- $\widehat{p}_k^{(i)} \in (0, 1)$: updated predicted probability for the k-th class of instance i with a value between 0 and 1.
- $\widehat{t}_k^{(i)} = \begin{cases} 1 & \text{if } \widehat{p}_k^{(i)} \geq \theta_k \\ 0 & \text{otherwise.} \end{cases}$: updated predicted label for the k-th class of instance i.
- K: number of categories (aka classes) in a multi-class, multi-label problem. For example, suppose that we have a dataset that is labeled for the presence of cats, dogs, and rabbits in any given image. If a given image $X^{(i)}$ has cats and dogs but not rabbits, then $Y^{(i)} = \{1, 1, 0\}$.
- N: Number of instances.
- $X^{(i)}$: Data for instance i.
- $Y^{(i)} = \left\{y_1^{(i)}, y_2^{(i)}, \ \ldots, y_K^{(i)}\right\}$: True label set, for instance i. For example, consider a dataset that is labeled for the presence of cats, dogs, and rabbits in any given instance. If a given instance $X^{(i)}$ has cats and dogs but not rabbits, then $Y^{(i)} = \{1, 1, 0\}$.
- $P^{(i)} = \left\{p_k^{(i)}\right\}_{k=1}^{K}$: Predicted probability set obtained in the output of the classifier $F(\cdot)$ representing the probability that each class k is present in the sample.
- $T^{(i)} = \left\{t_k^{(i)}\right\}_{k=1}^{K}$: predicted label set, for instance i.
- $\mathbb{X} = \left\{X^{(i)}\right\}_{i=1}^{N}$: Set of all instances.
- $\mathbb{Y} = \left\{Y^{(i)}\right\}_{i=1}^{N}$: Set of all true labels.
- $\mathbb{D} = \{\mathbb{X}, \mathbb{Y}\}$: Dataset containing all instances and all true labels.
- $l_k^{(i)} = \mathcal{L}\left(y_k^{(i)}, p_k^{(i)}\right)$: $\mathcal{L}(\cdot)$ is an arbitrary loss function (e.g., binary cross entropy) that takes the true label $y_k^{(i)}$ and predicted probability $p_k^{(i)}$ for class k and instance i and outputs the loss value $l_k^{(i)}$. We will refer to this as the "base loss function" throughout this paper.

- Loss(θ): Measured loss for all classes and instances. This value will be obtained using a modified version of the base loss function $\mathcal{L}(\cdot)$ (e.g., with added regularization, etc.).
- $\omega_k^{(i)}$: Estimated weight for k-th class c_k of instance i with respect to its parent class Γ_k.
- $\widehat{l}_k^{(i)} = \omega_k^{(i)}\ l_k^{(i)}$: updated loss for class k and instance i.

2.3.2 Problem Formulation

Let us define the multi-label classification problem as follows. Let $\mathbb{X} = \left\{X^{(i)}\right\}_{i=1}^{N}$ be the set of N chest radiograph images and $\mathbb{Y} = \left\{Y^{(i)}\right\}_{i=1}^{N}$ be their corresponding ground truth labels. In the context of chest radiograph interpretation, the label set C typically includes various thoracic abnormalities such as pneumothorax, consolidation, atelectasis, and cardiomegaly. The ground-truth labels for the dataset were provided by experienced radiologists who annotated each image with the corresponding abnormalities.

Given the set of disease classes $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, let us define a directed acyclic graph (DAG) $\mathcal{G} = \{C, \mathcal{E}\}$ representing the taxonomy of thoracic diseases, where $\mathcal{E}$ is the set of directed edges representing parent-child relationships between these classes. For each node $c_k \in C$, let Λ_k be the parent node of class c_k and denote $\mathcal{J}_k \subset C$ the set of child classes of class c_k in DAG $\mathcal{G}$.

Let $\omega_k^{(i)}$ be a scalar weight assigned to the class c_k of instance i with respect to its parent class Λ_k. In multi-label classification problems, each sample can have multiple labels simultaneously assigned to it; thus, the sigmoid function is utilized to predict the probabilities for each class being present in a given sample. The output of the final layer of the neural network, for instance i, is passed through a sigmoid function to generate a set of values between 0 and 1 corresponding to the label set C to obtain a set of K predicted probabilities $P^{(i)} = \left\{p_k^{(i)}\right\}_{k=1}^{K}$. These predicted probabilities, derived from the sigmoid activation function, can be interpreted as the probability that the input sample belongs to each class. Consequently, the loss function quantifies the similarity between predicted and true labels.

Let us denote $l_k = \mathcal{L}\left(p_k^{(i)}, y_k^{(i)}\right),\ k \in \{1, 2, \ldots, K\}$ where $\mathcal{L}(\cdot)$ is an arbitrary and appropriate single class loss function for the task (e.g., binary cross-entropy, Dice, etc.) that is used to calculate the difference between the predicted probability $p_k^{(i)}$ and the true class label $y_k^{(i)}$ for instance i and class k.

2.3.3 Label Taxonomy Structure

To exploit the inherent hierarchical relationships between thoracic abnormalities, the first step is to define a disease taxonomy that demonstrates different abnormalities interrelationships. In this taxonomy, diseases will be structured hierarchically, with higher levels representing broader disease categories and lower levels representing more nuanced distinctions between related diseases. For example, pleural effusion and pneumothorax can be classified as subcategories of pleural abnormalities, whereas atelectasis and consolidation can be classified under pulmonary opacity. This hierarchical structure enables the model to take advantage of the relationships between diseases to improve its classification performance.

In medical imaging, labels are frequently organized as trees or directed acyclic graphs (DAGs) to represent the hierarchical relationships between different classes of labels. For example, a DAG can be used to represent the human body's organs, with each node representing a different organ and the edges representing the relationships between organs (e.g., the liver is part of the abdominal cavity). Using a tree or DAG structure for labels in medical imaging has a number of advantages, including improved accuracy and interpretability of classification algorithms, which are essential for making sense of the vast amounts of data generated by medical imaging technologies. In medical imaging, hierarchies of labels are typically constructed by subject matter experts with a comprehensive understanding of human anatomy and physiology, such as radiologists. Construction of these hierarchies can be challenging and time-consuming because it requires in-depth knowledge of the subject matter and the ability to organize complex data into clean and intuitive structures.

A comprehensive label taxonomy for lung diseases was developed based on the taxonomies presented by Irvin [64] for the CheXpert dataset and Chen [95] for the PADCHEST [63] and the CXR arm of the prostate, lung, colorectal and ovarian (PLCO) [96] datasets. This unified taxonomical structure is designed to be applied to various chest radiography datasets. The developed taxonomy structure is depicted in Figure 2.1.

2.3.4 Approach 1: Conditional Predicted Probability

When computational resources are limited, this technique can be applied to test samples without the need to fine-tune the pre-trained, multi-label classification model. This adaptability ensures that the benefits of considering hierarchical relationships between labels can be realized in a wide range of practical scenarios, without imposing excessive computational requirements.

Directly updating the predicted probabilities presents potential benefits, including the following:

- **Simplicity:** Direct modification of predicted probabilities eliminates the need for substantial changes to the loss function, thus facilitating implementation.
- **Faster convergence:** In some cases, direct updates can accelerate convergence due to a more accurate representation of hierarchical relationships, thus reducing the overall training time.
- **Improved performance in specific scenarios:** Depending on the problem and dataset, direct updates may provide superior performance in certain circumstances, especially when incorporating class relationships into the loss function is challenging.
- **Easier calibration:** Direct modification of predicted probabilities can facilitate calibration of the model output to more closely match the true label distribution.

The proposed technique provides an easy way to improve the performance of existing pre-trained models during inference time by updating the value of the predicted logit for each class that was obtained at the last layer of the neural network based on the predicted logit of its corresponding parent class. The aim is to calculate the conditional predicted probability for each class k and instance i, taking into account the predicted probability of the parent class. We can formalize this by defining a new predicted probability for the k -th class (c_k) and instance i as follows.

$$\widehat{p}_k^{(i)} = \frac{1}{1 + \exp\left(-\left(q_k^{(i)} + \alpha_{k,j} q_j^{(i)}\right)\right)} \tag{2.3.1}$$

where $j = \Lambda_k$ is the index of the parent class of the k-th class, and $\alpha_{k,j}$ is the hyperparameter that controls the influence of different parent class logits on child class logits.

When $\alpha_{k,j} = 0$, there is no influence from the parent class c_j on the child class c_k. By carefully selecting appropriate hyperparameter values, this transfer learning-based technique can be employed to effectively adjust the predicted probabilities of each class, considering the hierarchical relationship between classes, and potentially improving classification accuracy.

Parameter Selection and Tuning

The selection of appropriate hyperparameters is crucial for the effectiveness of the proposed transfer learning-based technique. In this study, we employ a systematic approach to tune the hyperparameters

$\alpha_{k,j}$, which controls the dependency between the predicted probabilities of the child and parent classes. We utilize a grid search method along with cross-validation to determine the optimal values for these hyperparameters. The search space for both hyperparameters is defined based on preliminary experiments and domain knowledge, ensuring a balance between model complexity and predictive performance.

2.3.5 Approach 2: Conditional Loss

In a second approach, we propose a similar concept to the approach discussed in section 2.3.4, however, rather than directly updating the predicted probability of each class, we instead update the loss value of each class based on the loss values of its parent classes. In the previous approach, we directly updated the predicted probability so that it could be applied unsupervised to existing pre-trained models. Although this method is highly useful during inference time, it presents some challenges if we use it during the optimization phase of our classifier model. Among these disadvantages are the following.

- **Inconsistency with the optimization process:** Direct updating of predicted probabilities can misalign with the optimization procedure, which typically minimizes the loss function, potentially resulting in learning inconsistencies.

- **Difficulty in fine-tuning:** Direct updates can complicate fine-tuning the method's impact on the model, whereas adjusting the influence of various components is often simpler when updating the loss value through weighting factors or hyperparameters.

- **Potential overfitting:** Direct modification of predicted probabilities could inadvertently overfit the model to particular hierarchical relationships in the training data, thus hindering generalization to unseen data.

The utilization of the loss function based approach can prove advantageous in certain scenarios, particularly in the context of multi-label classification tasks that involve hierarchical relationships, as it offers numerous benefits.

- **Emphasis on error minimization:**The loss values represent the difference between the predictions made by the model and the actual labels provided as ground-truth. Incorporating parent class loss values into child class loss calculations aims to minimize errors throughout the hierarchy, thereby assuring accurate predictions for both parent and child classes.

- **Enhanced gradient propagation:** During the training of deep learning models, the model parameters are updated by backpropagating gradients through layers. Incorporating the loss values of the parent class through the calculation of the loss for the child class improves the connections between the parent and child classes with respect to the propagation of gradients. This may lead to more effective acquisition of hierarchical associations and expedited convergence in the course of training.

- **Robustness to label noise:** Real-world datasets may exhibit inconsistencies or noise in their ground truth labels. The inclusion of loss values from parent classes in the computation of loss values for child classes enhances the consistency of the hierarchy by penalizing deviations from anticipated parent-child associations. This approach improves the model's resilience to possible label inaccuracies in the dataset.

- **Improved interpretability:** Employing loss values instead of predicted probabilities facilitates a more straightforward comprehension of the model's ability to capture hierarchical interrelationships among classes. The impact of high loss values on parent classes is more pronounced on the losses of their corresponding child classes, indicating the necessity to improve specific areas to better reflect these associations.

Formulation of the Proposed Technique

In multi-label classification problems, where each sample may belong to multiple classes, it is often necessary to combine the loss values for all classes to effectively train the model. Various methods can be employed to achieve this, depending on the specific problem. A common approach is to calculate the average loss across all classes for each sample by summing the losses for each class of a given sample and dividing the sum by the total number of classes to which the sample belongs. This method is effective when all classes are independent, of equal importance, and warrant equal weight in the total loss calculation. For instance, in the case of cross-entropy loss, we have:

$$l_k = -\left(y_k^{(i)} \log(p_k^{(i)}) + (1 - y_k^{(i)}) \log(1 - p_k^{(i)})\right) \tag{2.3.2}$$

$$\text{Loss}(\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} l_k \tag{2.3.3}$$

In this formulation, the objective is to minimize the loss function with respect to the model parameters θ, resulting in an optimal set of parameters that produce accurate predictions for multi-label classification tasks. However, class independence and equal importance between different classes cannot always be assumed. Inclusion of a hierarchical penalty or regularization term in the loss function is one way to push the loss function to take the taxonomy into account when optimizing the model hyperparameters (weights and biases). We use a regularization term β_k to penalize the loss for class c_k for each instance i in which the probability that its parent class c_j exists in that instance is low. This can be represented mathematically by adding a hierarchical penalty term $H(c_k|c_j)$ for the class c_k with respect to its corresponding parent class c_j as follows.

$$\widehat{l}_k^{(i)} = l_k^{(i)} + \beta_k H\left(c_k|c_j\right) \tag{2.3.4}$$

where $c_j = \Lambda(c_k)$, and β_k is the hyperparameter that balances the contributions of the class k's own loss value and its parent classes' loss values.

There are multiple ways to define the hierarchical penalty. For example, we can define it as the loss value of the parent class $l_j = L\left(y_j^{(i)}, p_j^{(i)}\right)$ as follows.

$$H(k|j) = \mathcal{L}\left(y_j^{(i)}, p_j^{(i)}\right) \tag{2.3.5}$$

Another approach to incorporating the interdependence between different classes into the loss function is to apply the loss function $\mathcal{L}$ to the true label of the parent class and the predicted probability of the child class as follows.

$$H\left(k|j\right) = \mathcal{L}\left(y_j^{(i)}, p_k^{(i)}\right) \tag{2.3.6}$$

In both Equations (2.3.5) and (2.3.6) the penalization term encourages the model to correctly predict the corresponding parent class when predicting the child class, ensuring that the predicted label set adheres to the hierarchical structure. In the aforementioned approach, we assume a linear relationship between child and parent losses, which can simplify the optimization process. However, this may not always accurately capture the relationship between the parent-child classes, as the relationship may not always be linear. Furthermore, the impact of the parent's loss on the total loss could be less significant, particularly if the child's loss is considerably greater than the parent's loss.

To address this problem, we can modify the loss measurements presented in Equations (2.3.5) and (2.3.6) to be based on the multiplication of losses rather than their addition.

Multiplying losses allows for a more flexible relationship between the child and parent classes, as it can model both linear and nonlinear relationships. Furthermore, the parent's loss can have a more significant impact on the total loss, since it is multiplied by the child's loss, ensuring that the hierarchical relationships are better captured. To achieve this, we can define the new loss as follows.

$$\widehat{l}_k^{(i)} = l_k^{(i)} H\left(c_k | c_j\right) \tag{2.3.7}$$

where the hierarchical penalty term is defined as follows.

$$H(k|j) = \begin{cases} 1 & \text{otherwise.} \\ a_k l_j^{(i)} + \beta_k & c_j \text{ has a parent} \end{cases} \tag{2.3.8}$$

where c_j is the parent class of the c_k class, and l_j is the parent loss value, for instance i.

The modified loss function in Equation (2.3.7) aims to ensure that predictions adhere to hierarchical relationships between classes by penalizing deviations from these established relationships. By adjusting the parameters α_k and β_k, we can regulate the degree to which hierarchical information influences the learning process.

2.3.6 Updating Loss Values and Predicted Probabilities

In the previous section, we introduced a taxonomy-based loss function with the goal of improving the classification accuracy of multi-class problems. However, one of the main advantages of our proposed technique is that it enables efficient utilization of pre-trained models and leverages the existing knowledge, thus reducing the computational cost and training time associated with re-optimization. In this section, we illustrate how both of our proposed approaches can be seamlessly integrated into the existing classification framework without the necessity to re-run the optimization phase of our classifier (e.g., DenseNet121). This can be achieved by focusing on updating the loss values (approach 2 shown in section 2.3.5) and predicted probabilities (approach 1 shown in section 2.3.4) to incorporate the hierarchical relationships present in the taxonomy structure.

During a training phase of a classifier (e.g., DenseNet121), an optimization algorithm such as gradient descent is used to determine the predicted probabilities that minimize the loss across the entire dataset. However, this approach is only valid during the training phase and only shows the predicted probability with respect to the original loss values measured by the classifier.

In the following, we show how to calculate the updated predicted probabilities from their updated loss values obtained from Equation (2.3.7) without re-doing the optimization process. Let us assume that binary cross entropy is used for the choice of the loss function $\mathcal{L}(\cdot)$. Let us denote $\widehat{q}_k^{(i)}, \widehat{p}_k^{(i)}$ as the updated values for logit and predicted probability of class k and instance i after applying the proposed technique. As previously discussed, to calculate the predicted probabilities, we need to pass the logits $\widehat{q}_k^{(i)}$ into a sigmoid function as shown below:

$$\widehat{p}_k^{(i)} = \text{sigmoid}\left(\widehat{q}_k^{(i)}\right) = \frac{1}{1+\exp\left(-\widehat{q}_k^{(i)}\right)} \tag{2.3.9}$$

The sigmoid activation function maps any value to a number between zero and one. The gradient of the sigmoid function (shown below) provides the direction in which the predicted probability must be updated.

$$\text{sigmoid}'\left(\widehat{q}_k^{(i)}\right) = \frac{\partial \text{sigmoid}}{\partial q} = \text{sigmoid}\left(\widehat{q}_k^{(i)}\right)\left(1 - \text{sigmoid}\left(\widehat{q}_k^{(i)}\right)\right) = \widehat{p}_k^{(i)}\left(1-\widehat{p}_k^{(i)}\right) \tag{2.3.10}$$

The loss gradient gives us the direction in which the predicted probability needs to be updated to minimize the loss. The gradient of the binary cross-entropy loss will be as follows.

$$\frac{\partial \mathcal{L}\left(\widehat{p}_k^{(i)},\ y_k^{(i)}\right)}{\partial \widehat{p}} = \frac{y_k^{(i)}}{\widehat{p}_k^{(i)}} - \frac{1-y_k^{(i)}}{1-\widehat{p}_k^{(i)}} \tag{2.3.11}$$

where $y_k^{(i)}$ and $\widehat{p}_k^{(i)}$ are the true label and predicted probability, respectively, for instance i and class k.

In the following equations, we show how we can use the predicted probability, the gradient loss shown in Equation (2.3.11) and the derivative of the sigmoid function shown in Equation (2.3.10) to calculate the updated predicted probability.

$$\frac{\partial \mathcal{L}\left(p_k^{(i)},\ y_k^{(i)}\right)}{\partial p}\ \text{sigmoid}'\left(\widehat{q}_k^{(i)}\right) = \left(\frac{y_k^{(i)}}{\widehat{p}_k^{(i)}} - \frac{1-y_k^{(i)}}{1-\widehat{p}_k^{(i)}}\right)\widehat{p}_k^{(i)}\left(1-\widehat{p}_k^{(i)}\right) = y_k^{(i)} - \widehat{p}_k^{(i)} \tag{2.3.12}$$

Hence, we can conclude the following.

$$\hat{p}_k^{(i)} = \begin{cases} -\frac{\partial \mathcal{L}\left(p_k^{(i)}, y_k^{(i)}\right)}{\partial p} \text{ sigmoid}'\left(\widehat{q}_k^{(i)}\right) + 1 & y = 1 \\ -\frac{\partial \mathcal{L}\left(p_k^{(i)}, y_k^{(i)}\right)}{\partial p} \text{ sigmoid}'\left(\widehat{q}_k^{(i)}\right) & \text{otherwise.} \end{cases} \tag{2.3.13}$$

We would like to modify this equation so that it does not directly depend on the true value and instead rely on the gradient loss. If we simplify the loss gradient shown in Equation (2.3.11) we will have the following:

$$\frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} = \frac{y_k^{(i)}}{\widehat{p}_k^{(i)}} - \frac{1 - y_k^{(i)}}{1 - \widehat{p}_k^{(i)}} = \frac{y_k^{(i)} - \widehat{p}_k^{(i)}}{\widehat{p}_k^{(i)}(1 - \widehat{p}_k^{(i)})} \tag{2.3.14}$$

In this equation, we can see that when the true label is positive $\left(y_k^{(i)} = 1\right)$, the loss gradient can only be 0 or a positive number. Similarly, when $\left(y_k^{(i)} = 0\right)$, the loss gradient can only take the value 0 or a negative number. Thus, we can modify the Equation (2.3.13) to look as follows.

$$\widehat{p}_k^{(i)} = \begin{cases} -\frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} \text{ sigmoid}'(q_k^{(i)}) + 1 & \text{if } \frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} \geq 0 \\ -\frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} \text{ sigmoid}'(q_k^{(i)}) & \text{otherwise.} \end{cases} \tag{2.3.15}$$

Finally, the Equation (2.3.15) can be simplified as follows.

$$\widehat{p}_k^{(i)} = \begin{cases} \exp(-\widehat{l}_k^{(i)}) & \text{if } \frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} \geq 0 \\ 1 - \exp(-\widehat{l}_k^{(i)}) & \text{otherwise} \end{cases} \tag{2.3.16}$$

where, $\widehat{l}_k^{(i)}$ is the updated loss for class k and instance i.

The following demonstrates the Equation (2.3.16) based on predicted probability to demonstrate its similarity to Equation (2.3.1) in Approach 1 (section 2.3.4). From Equation (2.3.8) we have $\hat{l}_k^{(i)} = l_k^{(i)}\left(\alpha_k\, l_j^{(i)} + \beta_k\right)$. By substituting that into $\exp\left(-\widehat{l}_k^{(i)}\right)$, for $y_k^{(i)} = 1$ we would have the following equation.

$$\exp\left(-\widehat{l}_k^{(i)}\right) = \exp\left(-l_k^{(i)}\left(\alpha_k\, l_j^{(i)} + \beta_k\right)\right) = \left(p_k^{(i)}\right)^{-\alpha_k \log\left(p_j^{(i)}\right) + \beta_k} \tag{2.3.17}$$

Furthermore, $1-\exp\left(-\widehat{l}_k^{(i)}\right)$, for $y_k^{(i)}=0$ will be as follows.

$$1-\exp\left(-\widehat{l}_k^{(i)}\right)=1-\exp\left(-l_k^{(i)}\left(\alpha_k\ l_j^{(i)}+\beta_k\right)\right)=1-\left(1-p_k^{(i)}\right)^{-\alpha_k\log\left(1-p_j^{(i)}\right)+\beta_k} \tag{2.3.18}$$

By substituting the Equations (2.3.17) and (2.3.18) into Equation (2.3.16) we will have the following.

$$\widehat{p}_k^{(i)}=\begin{cases}\left(p_k^{(i)}\right)^{-\alpha_k\log(p_j^{(i)})+\beta_k} & \text{if} \quad y_k^{(i)}=1\\ 1-\left(1-p_k^{(i)}\right)^{-\alpha_k\log\left(1-p_j^{(i)}\right)+\beta_k} & \text{otherwise.}\end{cases} \tag{2.3.19}$$

2.3.7 Experimental Setup

Datasets

Three diverse and publicly available datasets are used to evaluate the proposed hierarchical multi-label classification techniques: CheXpert [64], PADCHEST [63], and VinDr-CXR [97]. These datasets contain a diverse range of chest radiographic images covering various thoracic diseases, providing a comprehensive evaluation of the effectiveness of our method. The description of the three datasets are as follows.

- **CheXpert** [64] is a large-scale dataset containing 224,316 chest radiographs of 65,240 patients, labeled with 14 radiographic findings.
- **PADCHEST** [63] consists of 160,000 chest radiographs of 67,000 patients, annotated with 174 radiographic findings. This dataset is highly diverse and includes a wide variety of thoracic diseases.
- **NIH** [62] includes 112,120 chest radiographs of 30,805 patients labeled with 14 categories of thoracic diseases.

Preprocessing: The chest radiographs were pre-processed to ensure consistency across the datasets. The images were resized to a resolution of 224×224 pixels, with the pixel intensities normalized to a range of 0 and 1. Data augmentation techniques, such as rotation, translation, and horizontal flipping, were applied to increase the dataset's size and diversity, consequently enhancing the model's generalization capability.

Model Optimization

The DenseNet121 [98] architecture and the pre-trained weights provided by Cohen [99] was used as the baseline model. The model was fine-tuned on a subset of CheXpert [64], NIH [62], PADCHEST [63] for 18 thoracic diseases. A series of transformations were applied to all train images, including rotation of up to 45 degrees, translation of up to 15%, and scaling up to 10%. Binary cross entropy losses and Adam optimizer were used.

Parallelization for multiple CPU cores: To effectively optimize the hyperparameters of our proposed taxonomy-based transfer learning methods, we utilize parallelization techniques that distribute the computational load across multiple CPU cores. By leveraging the power of parallel processing, we can drastically reduce the overall computation time and accelerate the optimization procedure, making the method more applicable to large-scale and high-dimensional datasets. Different parallelization libraries, such as joblib and Python multiprocessing, were employed to facilitate the implementation of parallelism, ensuring seamless integration with existing frameworks and offering a scalable and hardware-adaptable solution.

Optimum Threshold Determination: Determining the optimal threshold is a crucial aspect of evaluating the performance of the proposed method, as it determines the point at which the predictions for multi-label classification tasks are translated into binary class labels. To determine the optimal threshold value, we used receiver operating characteristic (ROC) analysis, a common method for evaluating the performance of classification models. ROC analysis provides a comprehensive view of the model's performance at various threshold values, allowing us to determine the optimal point for balancing the true positive rate (sensitivity) and the false positive rate (specificity) (1-specificity). By plotting the ROC curve and calculating the area under the curve (AUC), we can quantitatively evaluate the discriminatory ability of the model and compare its performance at various threshold values. The optimal threshold is determined by locating the point on the ROC curve closest to the upper left corner, which represents the highest true positive rate and the lowest false positive rate. By incorporating ROC analysis and optimal threshold determination into our experimental design, we ensure that our results not only accurately reflect the performance of the model but also provide valuable insight into the practical applicability of our approach in real-world settings.

Evaluation: To assess the performance of the proposed techniques in accurately classifying samples compared to a baseline model, several evaluation metrics were used. The metrics were selected based on their ability to provide a comprehensive assessment of the model's performance in terms of accuracy, precision, recall, and the ability to differentiate between true and false positives. The evaluated metrics

are as follows.

- **Accuracy** measures the proportion of correctly classified samples to the total number of samples.
- **F1-score** is the harmonic mean of precision and recall, providing a balanced assessment of the method's performance.
- **Area Under the Receiver Operating Characteristic Curve (AUROC)**: The ROC curve is a graphical representation of the diagnostic performance of a binary classifier system as its discrimination threshold is varied. The ROC curve is derived by plotting the true positive rate (TPR) versus the false positive rate (FPR) at different thresholds. The AUC provides a single scalar value representing the expected performance of the classifier. An AUC of 1 indicates that the classifier can distinguish perfectly between the two classes (e.g., "positive" and "negative"), whereas an AUC of 0.5 indicates that the classifier is no better than random chance.
- **t-stat (t-statistic)** is a measurement of the magnitude of the difference relative to the variance in sample data. The t-value quantifies the statistical significance of the difference. It is used to test hypotheses regarding the mean or the difference between two means when the standard deviation of the population is unknown.
- **p-value**: In hypothesis testing, the p-value is a function used to determine the significance of the results. It represents the probability that test results were generated at random. If the p-value is small (typically 0.05), there is strong evidence that the null hypothesis should be rejected.
- **Cohen's Kappa** measures the concordance between two raters who classify items into mutually exclusive categories. Primarily, it is used to determine the degree of agreement between two raters. The Kappa score takes into account the possibility that the agreement occurred by chance. A Kappa score of 1 indicates perfect concordance between two raters. A Kappa score of less than 1 indicates less than perfect agreement, and a Kappa score of less than 0 indicates either no agreement or agreement that is worse than random.
- **BF10 (Bayes Factor)** rates the strength of the evidence in favor of one statistical model over another, given the available data. BF10 specifically contrasts the evidence supporting a null hypothesis (H0) with an alternative hypothesis (H1). The data are equally likely to be true under the null and alternative hypotheses, according to a BF10 value of 1. A BF10 value greater than 1 denotes support for H1, while a value lower than 1 denotes support for H0. In general, values

between 1/3 and 3 are regarded as inconclusive, values above 3 as some evidence for H1, and values below 1/3 as evidence for H0.

- **Cohen's d** is a measure of effect size in the context of a t-test for the difference between two means. It can be calculated as the difference between two means divided by the data's standard deviation. Typically, small, medium, and large effect sizes are referred to as Cohen's d values of 0.2, 0.5, and 0.8, respectively. It is a common method of estimating the difference between two groups after adjusting for variance and sample size variations.

- **Power (Statistical Power)** is the likelihood that a test will correctly reject the null hypothesis when the alternative hypothesis is true (i.e., the test will not make a Type II error). Power is typically desired to be 0.8 or higher, meaning there is an 80% or greater chance of discovering a true effect if it is present. Many variables, such as the effect size, sample size, significance level, and data variability, can have an impact on power. Calculating power can be used to determine the sample size required to detect an effect of a given size when designing a study.

Some limitations of these metrics are as follows. While accuracy is a useful metric for evaluating overall performance, it may not be the most appropriate metric for unbalanced datasets in which the number of samples in each class is significantly different. Similarly, F1-score may be biased towards the class with a larger sample size, and AUROC may not be appropriate for datasets with a high degree of class overlap. In addition, outliers or non-normal distributions may influence the t-statistic and p-value, whereas Cohen's Kappa may not be applicable to non-categorical data. BF10 may be affected by the selection of prior probabilities, and Cohen's d may not apply to non-parametric data. The choice of significance level and the data variability may have an effect on the power.

2.4 Results

Figure 2.1 shows the created taxonomy structure. This comprehensive classification system accumulated using taxonomy graphs in Irvin [64], and Chen [95] helps categorize various disease manifestations observed in public datasets, such as CheXpert, PADCHEST and NIH and serves as a framework for understanding and analyzing chest radiograph abnormalities.

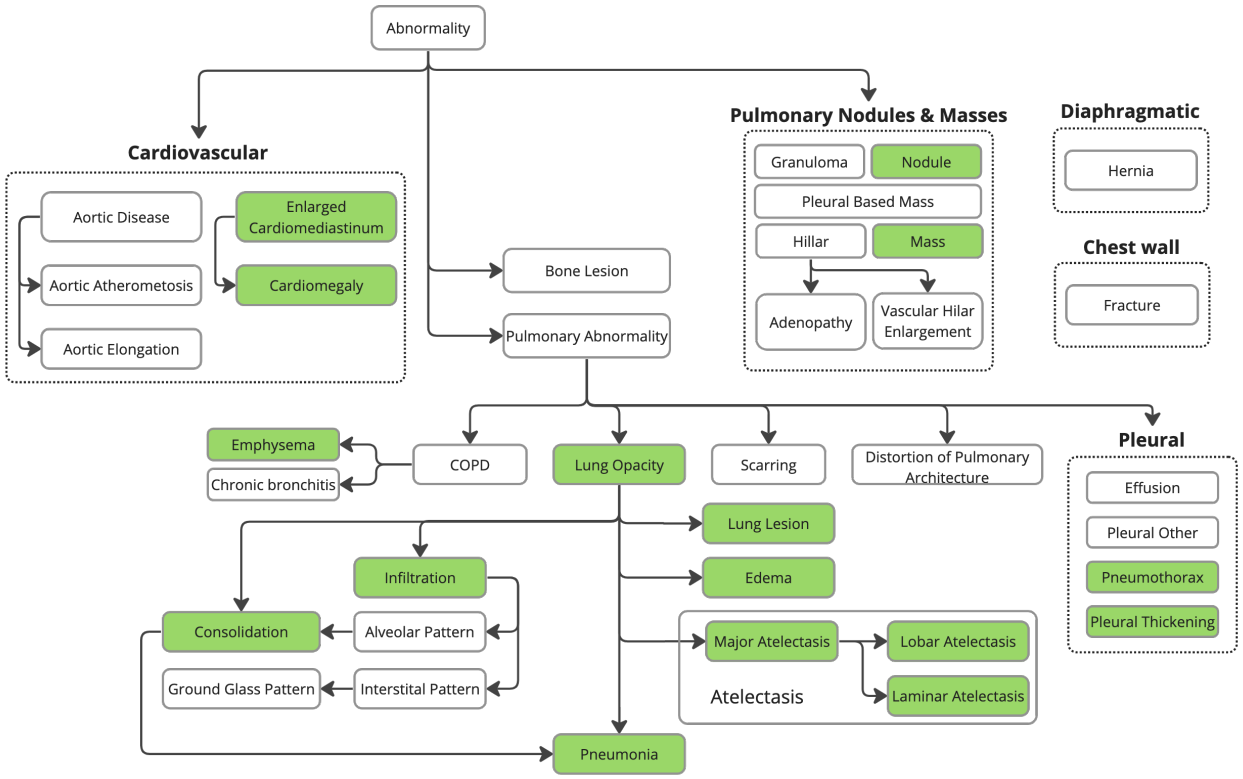

Figure 2.1: Taxonomy structure of lung pathologies in chest radiographs.

In this study, we investigated the frequency of various pathological labels in three distinct medical imaging datasets: CheX [64], PADCHEST [63], NIH [62]. Table 2.1 depicts the presence of each pathology label across these datasets. To conform to cohen [99] work, the same 18 pathologies selected by Cohen [99] are used for model optimization. Special consideration is given to the pathologies that appear in at least two of the three datasets and are included in our taxonomy. These pathologies are marked with a green color in the table and include **Atelectasis**, **Consolidation**, **Infiltration**, **Edema**, **Pneumonia**, **Cardiomegaly**, **Lung Lesion**, **Lung Opacity**, **Enlarged Cardiomediastinum**. This selection is significant because it reflects the pathologies that not only manifest more frequently across multiple data sources, but are also reflected in the hierarchical taxonomy used in our study. The study's analysis and conclusions are focused on these specific pathologies due to their consistent presence and relevance within the taxonomy structure. Further, the cross-dataset presence of these pathologies enhances the generalizability of our study, as the developed models are validated on multiple independent datasets. The pathologies that are not highlighted, i.e., those occurring in just one or none of the datasets or not included in our taxonomy, were not included in the final evaluation of this study. Their exclusion is mainly due to the lack of sufficient data for a robust comparison or their non-alignment with the

studied taxonomy structure.

Table 2.1: Pathologies present in each dataset

Pathologies	NIH	PADCHEST	CheX
Air Trapping		X	
Aortic Atheromatosis		X	
Aortic Elongation		X	
Aortic Enlargement			
Atelectasis	X	X	X
Bronchiectasis		X	
Calcification			
Calcified Granuloma			
Cardiomegaly	X	X	X
Consolidation		X	X
Costophrenic Angle Blunting		X	
Edema	X	X	X
Effusion	X	X	X
Emphysema	X	X	
Enlarged Cardiomediastinum			X
Fibrosis	X	X	
Flattened Diaphragm		X	
Fracture		X	X
Granuloma		X	

Pathologies	NIH	PADCHEST	CheX
Hemidiaphragm Elevation		X	
Hernia	X	X	
Hilar Enlargement		X	
ILD			
Infiltration	X	X	
Lung Lesion			X
Lung Opacity			X
Mass	X	X	
Nodule/Mass			
Nodule	X	X	
Pleural Other			X
Pleural Thickening	X	X	
Pneumonia	X	X	X
Pneumothorax	X	X	X
Pulmonary Fibrosis			
Scoliosis		X	
Tuberculosis		X	
Tube		X	

Table 2.2 shows the number of instances that has a specific pathology in each of the three studied datasets (CheX [64], PADCHEST [63], NIH [62]). Prior to applying the proposed technique a set of preprocessing steps are applied to ground truth label set. In medical images with multiple classes, it is common for the labeler to only label the pathologies that their study requires. This sometimes result in situations where some instances of data are labeled for the presence of some of the child pathologies but not their corresponding parent pathologies. To compensate for this lack of labeling for some parent classes which is necessary for the effectiveness of the proposed techniques, we updated the label value indicating the presence of classes with at least one child class to TRUE (indicating the class exist in that instance). This preprocessing is applied to all pathologies which are not labeled in the original ground truth label set. As can be seen in Table 2.2 (highlighted cells), while the Lung Opacity and Enlarged Cardiomediastinum classes were not present in the original ground truth label

sets of NIH and PADCHEST datasets (Table 2.1), by updating the ground truth label set we end up with multiple instances where based on the presence of their child classes' presence we have determined the presence of the respective parent class.

Table 2.2: Number of samples present in the evaluated datasets (CheX, NIH, and PC) per pathology.

Pathologies\Dataset	CheXpert		NIH		PADCHEST	
	PA	AP	PA	AP	PA	AP
Atelectasis	2460	11643	1557	1016	2419	232
Consolidation	1125	4956	384	253	475	77
Infiltration	0	0	3273	1131	4309	587
Pneumothorax	1060	4239	243	253	97	15
Edema	1330	15117	39	237	108	130
Emphysema	0	0	264	193	546	30
Fibrosis	0	0	556	61	341	8
Effusion	5206	19349	1269	654	1625	311
Pneumonia	992	2064	175	89	1910	211
Pleural_Thickening	0	0	745	145	2075	34
Cardiomegaly	2117	8284	729	203	5387	261
Nodule	0	0	1609	460	2190	95
Mass	0	0	1213	493	506	17
Hernia	0	0	81	13	988	38
Lung Lesion	1655	3110	0	0	0	0
Fracture	1115	3463	0	0	1662	69
Lung Opacity	7006	28183	4917	2216	6947	861
Enlarged Cardiomediastinum	1100	4577	729	203	5387	261
Total	20543	53359	28868	9060	61692	2445

Figure 2.2 presents the comparison of the performance of our proposed techniques "logit" and "loss" against the "baseline" technique for a series of nine medical conditions related to lung and heart diseases on three datasets (CheXpert, PADCHEST, NIH). These nine pathologies include the two parent classes (**Lung Opacity**, and **Enlarged Cardiomediastinum**) and their corresponding child classes, as

shown in Figure 2.1. The individual subplots exhibit overlaid receiver operating characteristic (ROC) curves and their corresponding AUC scores.

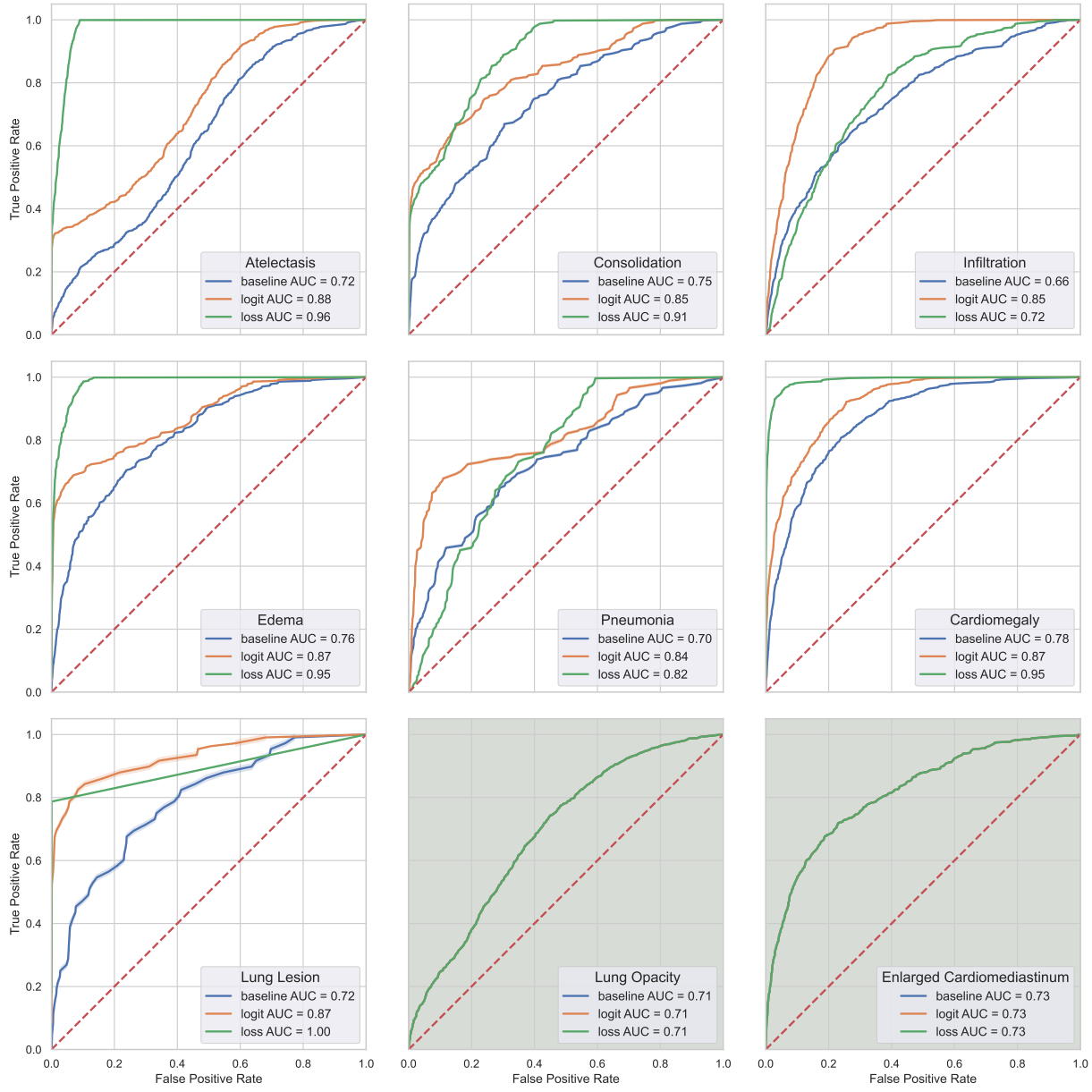

Figure 2.2: Comparative analysis of the ROC curves for nine thoracic pathologies using the "logit" and "loss" techniques as well as the baseline. The subplots highlighted with a darker background, represent parent class diseases.

Table 2.3 presents the comparison of the performance of our proposed techniques "logit" and "loss"

against the “baseline” technique for various statistical metrics. The “logit” method (upper table), shows a considerable improvement over the “baseline” across all conditions. We observe kappa values between 0.495 and 1. The kappa statistic is a measure of agreement between two methods, with a value of 1 indicating perfect agreement. The p-value for all child classes show a value less than 0.05 (ranging from 2.1E-89 to 2.9E-16), which indicates the statistically significant superiority of the “logit” method over the “Baseline”. High t-statistics and power values of 1 further confirm the robustness of our method. In terms of Bayes factor, results for the “logit” method are extremely strong for all conditions suggesting a high amount of evidence in favor of the “logit” method for these conditions. The second proposed method, “loss”, also presents promising results when compared to the “baseline”, but with more variation. Kappa values ranged from a low of 0.059 for Lung Lesion to a high of 0.836 for Infiltration. The p-values indicate statistically significant differences for most conditions, although Infiltration and Pneumonia present p-values greater than 0.05 (0.053 and 0.207 respectively), suggesting that the performance improvement over the “baseline” for these conditions may not be statistically significant. T-statistics are high, and power values are 1 for all conditions except Infiltration and Pneumonia. The cohen-d values for this method are generally larger than those in the “logit” method, indicating a larger effect size. In terms of Bayes factor, results for the “loss” method are extremely strong for conditions such as Atelectasis and Edema, while being much lower for conditions like Infiltration and Pneumonia, suggesting a reduced amount of evidence in favor of the “loss” method for these conditions. Both the “logit” and “loss” methods show significant improvements over the “baseline” method in the diagnosis of several heart and lung conditions, albeit with some variance in the degree of improvement. While the “logit” method demonstrates a more consistent level of improvement across all conditions, the “loss” method shows potential for even greater improvement in certain conditions but with less consistency across the conditions studied.

Table 2.3: Statistical performance comparison between the proposed techniques "logit" and "loss" and the "baseline" technique across various pathologies. The upper table displays the findings of the "logit" technique, while the lower table displays the findings of the "loss" technique. The reported metrics for each pathology are the Kappa statistic, p-value, t-statistic, statistical power, Cohen's d, and Bayes Factor (BF10). A kappa value of 1 indicates perfect agreement between techniques, whereas a larger Bayes factor indicates greater support for the "logit" or "loss" technique over the baseline.

		kappa	p_value	t_stat	power	cohen-d	BF10
L	Atelectasis	0.495	2.1E-89	20.2	1	0.346	3.0E+85
	Consolidation	0.508	2.0E-18	8.8	1	0.150	8.3E+14
O	Infiltration	0.620	2.7E-28	11.1	1	0.190	4.9E+24
	Edema	0.614	1.2E-52	15.3	1	0.263	7.2E+48
G	Pneumonia	0.573	2.9E-16	8.2	1	0.140	6.3E+12
	Cardiomegaly	0.615	1.9E-72	18.1	1	0.310	3.9E+68
I	Lung Lesion	0.580	7.0E-23	9.9	1	0.169	2.1E+19
	Lung Opacity	1	1	0	0.05	0	0.019
T	Enlarged Cardiomediastinum	1	1	0	0.05	0	0.019

		kappa	p_value	t_stat	power	cohen-d	BF10
	Atelectasis	0.222	4.9E-183	29.3	1	0.502	7.7E+178
L	Consolidation	0.310	4.3E-116	23.1	1	0.396	1.2E+112
	Infiltration	0.836	0.053	1.9	0.49	0.033	0.125
O	Edema	0.343	4.4E-190	29.9	1	0.512	8.2E+185
	Pneumonia	0.394	0.207	1.3	0.24	0.022	0.043
S	Cardiomegaly	0.501	1.2E-101	21.6	1	0.370	4.7E+97
	Lung Lesion	0.059	1.2E-207	31.3	1	0.537	2.9E+203
S	Lung Opacity	1	1	0	0.05	0	0.019
	Enlarged Cardiomediastinum	1	1	0	0.05	0	0.019

Figure 2.3 compares the performance of the proposed "loss" and "logit" techniques to the "baseline"

across three key metrics: accuracy (ACC), area under the receiver operating characteristic curve (AUC), and F1 score for various pathologies. The accuracy metric presents a clear advantage for the "loss" and "logit" methods over the "baseline" for the child classes of pathologies, a pattern that is consistent with the kappa statistics presented earlier. For instance, in Atelectasis, the "loss" method has an accuracy of 0.922 compared to 0.686 for the "baseline", while the "logit" method stands at 0.874. As expected, the parent classes, lung opacity and enlarged cardiomediastinum, show no difference between the techniques, with an accuracy of 0.663 and 0.696, respectively. The AUC, a model performance metric that accounts for both sensitivity and specificity, demonstrates once more that the "loss" and "logit" methods for the child classes are superior. For instance, in the case of cardiomegaly, the AUC is improved by 21% and 11% using the loss and logit methods, respectively. The AUC values for lung opacity and an enlarged cardiomediastinum, the parent classes, are identical for all three methods. The F1 score, which is the harmonic mean of precision and recall, sheds additional light on the enhanced performance of our proposed techniques. Notably, lung lesion increases from 0.094 in the "baseline" method to 0.982 in the "loss" method and 0.263 in the "logit" method. These results further validate our earlier findings that the "logit" and "loss" methods provide significant performance improvements over the "baseline" method for the majority of the child classes. Across all metrics and conditions, the "loss" method appears to perform marginally better than the "logit" method.

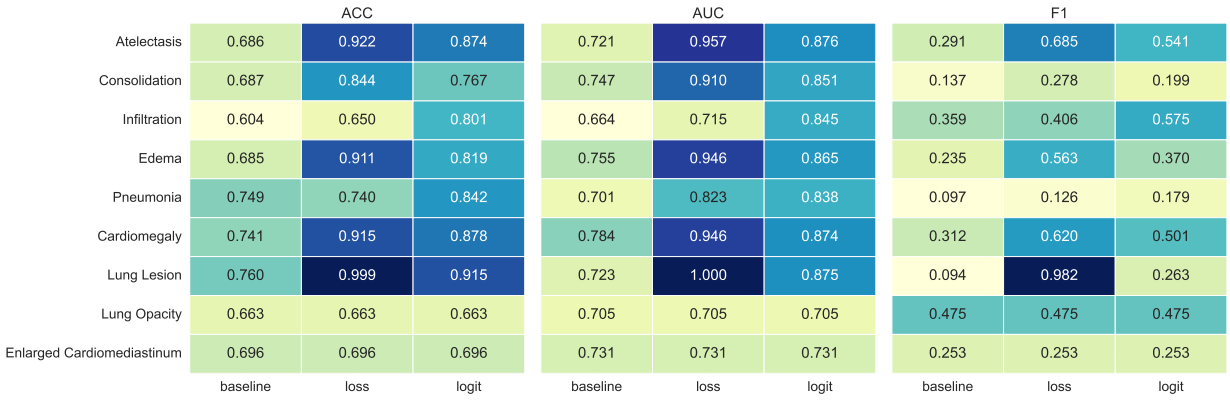

Figure 2.3: Heatmap visualization of model performance metrics across all three datasets. The subplots from left to right correspond to the Accuracy (ACC), Area Under the ROC Curve (AUC), and F1 Score for the baseline, "loss", and "logit" techniques respectively. The pathologies are shared on the y-axis. Darker colors signify higher values, indicating better model performance. Each cell represents the value of the corresponding metric for the given technique on a specific pathology

2.5 Discussion and Conclusion

In this work, we presented two hierarchical multi-label classification methods aimed at enhancing thoracic disease diagnosis in chest radiography. The first approach, denoted as the "loss" method, refines the loss value for each pathology, factoring in the influence of parent pathologies in the hierarchical structure. The "logit" method is a powerful tool for improving the performance of pre-trained models. This approach updates the logit values of each pathology based on the corresponding logit values of their parent pathologies, leveraging the inherent taxonomical relationships between pathologies. This strategy is particularly useful when ground truth labels are not readily available, as it allows for efficient enhancement of model performance without requiring access to such labels. By avoiding the need to build new models from scratch, the "logit" method offers a streamlined and effective approach to optimizing model performance in a variety of contexts. Whether working with complex medical data or other types of information, this approach can help researchers and practitioners achieve more accurate and reliable results with greater ease and efficiency.

The results of the study demonstrate the effectiveness of proposed hierarchical multi-label classification methodologies in improving the precision of thoracic disease diagnosis. Various performance metrics, including accuracy, AUC, and F1 scores, as well as Cohen's d, Cohen's kappa, t-statistics, p-value, and Bayes factor, are used to evaluate the performance of the proposed techniques against the baseline on three public datasets (CheXpert [64], PADCHEST [63], and NIH [62]), showing substantial improvements in the proposed techniques against the baseline. These findings suggest that these methods can be used as reliable tools for accurate and efficient diagnosis of thoracic diseases. Further research is needed to explore the potential benefits of these methods in clinical practice.

The utilization of logit adjustments represents a simple yet powerful approach for incorporating label hierarchy into a model without requiring significant modifications to the existing framework. However, this approach has the potential to obscure the optimization and learning process and results. On the other hand, modifying loss values is more closely aligned with the optimization process of the model, enabling a more refined adjustment of the hierarchical influence through weighting factors. This methodology promotes resilience to inaccuracies in labeling and improves conformity with established hierarchical relationships.

In essence, both the "loss" and "logit" techniques effectively leverage disease taxonomy to bolster classification performance, reinforcing the significance of exploiting label relationships in classification tasks. Moreover, these hierarchical techniques can potentially aid clinicians by improving the inter-

pretability of the models' predictions. Exploring predictions at varying levels of granularity based on taxonomy could facilitate personalized diagnoses tailored to individual clinical needs. Additionally, the techniques could be integrated into computer-aided diagnosis systems to provide more accurate and efficient diagnoses, potentially reducing the workload of clinicians and improving patient outcomes.

However, these methodologies exhibit certain constraints. Extending these methodologies to other applications would necessitate the development of a taxonomical structure for the labels of the corresponding dataset. The construction of such a structure may present difficulties for complex applications and typically necessitates the consensus of multiple domain experts. Additionally, the performance of the proposed techniques may be influenced by the quality and consistency of the labeling in the datasets, which may vary across different sources. Future studies should aim to evaluate the techniques on a broader range of datasets and consider the impact of labeling quality on performance.

Appendices

Acknowledgements

References

[1] F. Tao, L. Jiang, and C. Li, "Label Similarity-Based Weighted Soft Majority Voting and Pairing for Crowdsourcing," *Knowl Inf Syst*, vol. 62, pp. 2521–2538, July 2020.

[2] L. Jiang, G. Kong, and C. Li, "Wrapper Framework for Test-Cost-Sensitive Feature Selection," *IEEE Trans. Syst. Man Cybern, Syst.*, pp. 1–10, 2019.

[3] L. Jiang, L. Zhang, L. Yu, and D. Wang, "Class Specific Attribute Weighted Naive Bayes," *Pattern Recognition*, vol. 88, pp. 321–330, Apr. 2019.

[4] T. Tian, J. Zhu, and Y. Qiaoben, "Max-Margin Majority Voting for Learning From Crowds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, pp. 2480–2494, Oct. 2019.

[5] C. Li, V. S. Sheng, L. Jiang, and H. Li, "Noise Filtering to Improve Data and Model Quality for Crowdsourcing," *Knowledge-Based Systems*, vol. 107, pp. 96–103, Sept. 2016.

[6] C. Li, L. Jiang, and W. Xu, "Noise Correction to Improve Data and Model Quality for Crowdsourcing," *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 184–191, June 2019.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Conf. Comput. Vis. Pattern Recognit.*, (Miami, FL), pp. 248–255, IEEE, June 2009.

[8] Q. Liu, J. Peng, and A. T. Ihler, "Variational Inference for Crowdsourcing," in *Adv. Neural Inf. Process. Syst.*, vol. 25, Curran Associates, Inc., 2012.

[9] D. R. Karger, S. Oh, and D. Shah, "Budget Optimal Task Allocation for Reliable Crowdsourcing Systems," *Operations Research*, vol. 62, pp. 1–24, Feb. 2014.

[10] A. Sheshadri and M. Lease, "SQUARE: A Benchmark for Research on Computing Crowd Consensus," *HCOMP*, vol. 1, pp. 156–164, Nov. 2013.

[11] J. Tu, G. Yu, C. Domeniconi, J. Wang, G. Xiao, and M. Guo, "Multi-Label Answer Aggregation Based

on Joint Matrix Factorization," in *2018 IEEE Int. Conf. Data Min. ICDM*, (Singapore), pp. 517–526, IEEE, Nov. 2018.

[12] J. Zhang and X. Wu, "Multi-Label Inference for Crowdsourcing," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, (London United Kingdom), pp. 2738–2747, ACM, July 2018.

[13] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, L. Moy, C. Raykar, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning From Crowds," *JMLR*, vol. 11, no. 43, pp. 1297–1322, 2010.

[14] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth Inference in Crowdsourcing: Is the Problem Solved?," *Proc. VLDB Endow.*, vol. 10, pp. 541–552, Jan. 2017.

[15] A. P. Dawid and A. M. Skene, "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm," *Applied Statistics*, vol. 28, no. 1, p. 20, 1979.

[16] J. Liu, F. Tang, L. Chen, and Y. Zhu, "Exploiting Predicted Answer in Label Aggregation to Make Better Use of the Crowd Wisdom," *Information Sciences*, vol. 574, pp. 66–83, Oct. 2021.

[17] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced Data Management: Overview and Challenges," in *Proc. 2017 ACM Int. Conf. Manag. Data*, (Chicago Illinois USA), pp. 1711–1716, ACM, May 2017.

[18] M. Liu, L. Jiang, J. Liu, X. Wang, J. Zhu, and S. Liu, "Improving Learning-From-Crowds Through Expert Validation," in *Proc. 26th Int. Jt. Conf. Artif. Intell.*, (Melbourne, Australia), pp. 2329–2336, Aug. 2017.

[19] J. Zhang, V. S. Sheng, and J. Wu, "Crowdsourced Label Aggregation Using Bilayer Collaborative Clustering," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 30, pp. 3172–3185, Oct. 2019.

[20] W. Bi, L. Wang, J. T. Kwok, and Z. Tu, "Learning to Predict From Crowdsourced Data," in *Proc. 13th Conf. Uncertain. Artif. Intell.*, UAI'14, (Arlington, Virginia, USA), pp. 82–91, AUAI Press, July 2014.

[21] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: Leveraging Probabilistic Reasoning and Crowdsourcing Techniques for Large-Scale Entity Linking," in *Proc. 21st Int. Conf. World Wide Web*, (Lyon France), pp. 469–478, ACM, Apr. 2012.

[22] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan, "Spectral Methods Meet EM: A Provably Optimal

Algorithm for Crowdsourcing," in *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 1260–1268, Curran Associates, Inc., 2014.

[23] A. Kurve, D. J. Miller, and G. Kesidis, "Multi-Category Crowdsourcing Accounting for Variable Task Difficulty, Worker Skill, and Worker Intention," *IEEE Trans. Knowl. Data Eng.*, vol. 27, pp. 794–809, Mar. 2015.

[24] J. Zhang, X. Wu, and V. Sheng, "Imbalanced Multiple Noisy Labeling for Supervised Learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 27, pp. 1651–1652, June 2013.

[25] J. Hernandez-Gonzalez, I. Inza, and J. A. Lozano, "A Note on the Behavior of Majority Voting in Multi-Class Domains With Biased Annotators," *IEEE Trans. Knowl. Data Eng.*, vol. 31, pp. 195–200, Jan. 2019.

[26] P. Welinder, S. Branson, P. Perona, and S. Belongie, "The Multidimensional Wisdom of Crowds," in *Adv. Neural Inf. Process. Syst.*, vol. 23, Curran Associates, Inc., 2010.

[27] Y. Ma, A. Olshevsky, V. Saligrama, and C. Szepesvari, "Gradient Descent for Sparse Rank-One Matrix Completion for Crowd-Sourced Aggregation of Sparsely Interacting Workers," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5245–5280, 2020.

[28] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi, "Aggregating Crowdsourced Binary Ratings," in *Proc. 22nd Int. Conf. World Wide Web*, WWW '13, (New York, NY, USA), pp. 285–294, Association for Computing Machinery, May 2013.

[29] A. Ghosh, S. Kale, and P. McAfee, "Who Moderates the Moderators?: Crowdsourcing Abuse Detection in User-Generated Content," in *Proc. 12th ACM Conf. Electron. Commer.*, (San Jose, California, USA), p. 167, ACM Press, 2011.

[30] S. Warfield, K. Zou, and W. Wells, "Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation," *IEEE Trans. Med. Imaging*, vol. 23, pp. 903–921, July 2004.

[31] S. Winzeck, A. Hakim, R. McKinley, J. A. A. D. S. R. Pinto, V. Alves, C. Silva, M. Pisov, E. Krivov, M. Belyaev, M. Monteiro, A. Oliveira, Y. Choi, M. C. Paik, Y. Kwon, H. Lee, B. J. Kim, J.-H. Won, M. Islam, H. Ren, D. Robben, P. Suetens, E. Gong, Y. Niu, J. Xu, J. M. Pauly, C. Lucas, M. P. Heinrich, L. C. Rivera, L. S. Castillo, L. A. Daza, A. L. Beers, P. Arbelaezs, O. Maier, K. Chang, J. M. Brown, J. Kalpathy-Cramer,

G. Zaharchuk, R. Wiest, and M. Reyes, "ISLES 2016 and 2017-Benchmarking Ischemic Stroke Lesion Outcome Prediction Based on Multispectral MRI," *Front. Neurol.*, vol. 9, p. 679, Sept. 2018.

[32] O. Commowick, A. Istace, M. Kain, B. Laurent, F. Leray, M. Simon, S. C. Pop, P. Girard, R. Améli, J.-C. Ferré, A. Kerbrat, T. Tourdias, F. Cervenansky, T. Glatard, J. Beaumont, S. Doyle, F. Forbes, J. Knight, A. Khademi, A. Mahbod, C. Wang, R. McKinley, F. Wagner, J. Muschelli, E. Sweeney, E. Roura, X. Lladó, M. M. Santos, W. P. Santos, A. G. Silva-Filho, X. Tomas-Fernandez, H. Urien, I. Bloch, S. Valverde, M. Cabezas, F. J. Vera-Olmos, N. Malpica, C. Guttmann, S. Vukusic, G. Edan, M. Dojat, M. Styner, S. K. Warfield, F. Cotton, and C. Barillot, "Objective Evaluation of Multiple Sclerosis Lesion Segmentation Using a Data Management and Processing Infrastructure," *Sci Rep*, vol. 8, p. 13650, Sept. 2018.

[33] A. J. Asman and B. A. Landman, "Robust Statistical Label Fusion Through Consensus Level, Labeler Accuracy, and Truth Estimation (COLLATE)," *IEEE Trans. Med. Imaging*, vol. 30, pp. 1779–1794, Oct. 2011.

[34] A. J. Asman and B. A. Landman, "Formulating Spatially Varying Performance in the Statistical Fusion Framework," *IEEE Trans. Med. Imaging*, vol. 31, pp. 1326–1336, June 2012.

[35] J. Eugenio Iglesias, M. Rory Sabuncu, and K. Van Leemput, "A Unified Framework for Cross-Modality Multi-Atlas Segmentation of Brain Mri," *Medical Image Analysis*, vol. 17, pp. 1181–1191, Dec. 2013.

[36] M. Jorge Cardoso, K. Leung, M. Modat, S. Keihaninejad, D. Cash, J. Barnes, N. C. Fox, and S. Ourselin, "STEPS: Similarity and Truth Estimation for Propagated Segmentations and Its Application to Hippocampal Segmentation and Brain Parcelation," *Medical Image Analysis*, vol. 17, pp. 671–684, Aug. 2013.

[37] A. J. Asman and B. A. Landman, "Non-Local Statistical Label Fusion for Multi-Atlas Segmentation," *Benchmarking Ischemic Stroke Lesion*, vol. 17, pp. 194–208, Feb. 2013.

[38] A. Akhondi-Asl, L. Hoyte, M. E. Lockhart, and S. K. Warfield, "A Logarithmic Opinion Pool Based Staple Algorithm for the Fusion of Segmentations With Associated Reliability Weights," *IEEE Trans. Med. Imaging*, vol. 33, pp. 1997–2009, Oct. 2014.

[39] R. Artstein, "Inter-Annotator Agreement," in *Handbook of Linguistic Annotation* (N. Ide and J. Pustejovsky, eds.), pp. 297–313, Dordrecht: Springer Netherlands, 2017.

[40] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*. Los Angeles: SAGE, fourth

edition ed., 2018.

[41] J. Carletta, "Assessing Agreement on Classification Tasks: The Kappa Statistic," *Comput. Linguist.*, vol. 22, no. 2, pp. 249–254, 1996.

[42] V. S. Sheng, J. Zhang, B. Gu, and X. Wu, "Majority Voting and Pairing With Multiple Noisy Labeling," *IEEE Trans. Knowl. Data Eng.*, vol. 31, pp. 1355–1368, July 2019.

[43] J. Li, Y. Baba, and H. Kashima, "Incorporating Worker Similarity for Label Aggregation in Crowdsourcing," in *Artificial Neural Networks and Machine Learning (ICANN)*, vol. 11140 of *Lecture Notes in Computer Science*, (Cham), pp. 596–606, Springer International Publishing, 2018.

[44] V. Vapnik, "Principles of Risk Minimization for Learning Theory," in *Adv. Neural Inf. Process. Syst.*, vol. 4, Morgan-Kaufmann, 1991.

[45] M. Ayhan and P. Berens, "Test-Time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks," in *1st Conference on Medical Imaging with Deep Learning*, 2018.

[46] Z.-H. Zhou, "Ensemble Learning," *Encyclopedia of Biometrics*, pp. 270–273, 2009.

[47] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, pp. 1050–1059, PMLR, June 2016.

[48] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian Model Averaging: A Tutorial (With Comments by M. Clyde, David Draper and E. I. George, and a Rejoinder by the Authors," *Statist. Sci.*, vol. 14, Nov. 1999.

[49] V. Mullachery, A. Khera, and A. Husain, "Bayesian Neural Networks," 2018.

[50] X. Wang, D. Kondratyuk, E. Christiansen, K. M. Kitani, Y. Alon, and E. Eban, "Wisdom of Committees: An Overlooked Approach to Faster and More Accurate Models," 2020.

[51] A. N. Angelopoulos and S. Bates, "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification," 2021.

[52] D. Ustalov, N. Pavlichenko, and B. Tseitlin, "Learning From Crowds With Crowd-Kit," 2021.

[53] Q. Ma and A. Olshevsky, "Adversarial Crowdsourcing Through Robust Rank-One Matrix Completion," in *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 21841–21852, Curran Associates, Inc., 2020.

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

[55] D. Duan and C. Graff, "UCI Machine Learning Repository," 2017.

[56] crowd-kit, "Calculating Worker Agreement with Aggregate (Wawa)."

[57] D. Hovy, T. Berg-Kirkpatrick, A. Vaswani, and E. Hovy, "Learning Whom to Trust with MACE," in *North American Chapter of the Association for Computational Linguistics*, June 2013.

[58] N. Bellaviti, F. Bini, L. Pennacchi, G. Pepe, B. Bodini, R. Ceriani, C. D'Urbano, and A. Vaghi, "Increased Incidence of Spontaneous Pneumothorax in Very Young People: Observations and Treatment," *CHEST*, vol. 150, p. 560A, Oct. 2016.

[59] L. Delrue, R. Gosselin, B. Ilsen, A. Van Landeghem, J. de Mey, and P. Duyck, "Difficulties in the Interpretation of Chest Radiography," in *Comparative Interpretation of CT and Standard Radiography of the Chest* (E. E. Coche, B. Ghaye, J. de Mey, and P. Duyck, eds.), Medical Radiology, pp. 27–49, Berlin, Heidelberg: Springer, 2011.

[60] N. Crisp and L. Chen, "Global Supply of Health Professionals," *N Engl J Med*, vol. 370, pp. 950–957, Mar. 2014.

[61] J. Silverstein, "Most of the World Doesn't Have Access to X-Rays," *The Atlantic*, Sept. 2016.

[62] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," in *2017 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, (Honolulu, HI), pp. 3462–3471, IEEE, July 2017.

[63] A. Bustos, A. Pertusa, J.-M. Salinas, and M. de la Iglesia-Vayá, "Padchest: A Large Chest X-Ray Image Dataset With Multi-Label Annotated Reports," *Medical Image Analysis*, vol. 66, p. 101797, Dec. 2020.

[64] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P.

Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, "CheXpert: A Large Chest Radiograph Dataset With Uncertainty Labels and Expert Comparison," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 590–597, July 2019.

[65] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial Transformer Networks," in *Adv. Neural Inf. Process. Syst.* (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, Curran Associates, Inc., 2015.

[66] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A Survey on Deep Learning in Medical Image Analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, Dec. 2017.

[67] M. Eshghali, D. Kannan, N. Salmanzadeh-Meydani, and A. M. Esmaieeli Sikaroudi, "Machine Learning Based Integrated Scheduling and Rescheduling for Elective and Emergency Patients in the Operating Theatre," *Ann Oper Res*, Jan. 2023.

[68] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. P. C. Rodrigues, "Identifying Pneumonia in Chest X-Rays: A Deep Learning Approach," *Measurement*, vol. 145, pp. 511–518, Oct. 2019.

[69] P. Lakhani and B. Sundaram, "Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks," *Radiology*, vol. 284, pp. 574–582, Aug. 2017.

[70] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization," *Sci Rep*, vol. 9, p. 6268, Dec. 2019.

[71] W. Ausawalaithong, A. Thirach, S. Marukatat, and T. Wilaiprasitporn, "Automatic Lung Cancer Prediction From Chest X-Ray Images Using the Deep Learning Approach," in *11th Biomed. Eng. Int. Conf. BMEiCON*, (Chiang Mai), pp. 1–5, IEEE, Nov. 2018.

[72] G. Tsoumakas and I. Katakis, "Multi-Label Classification: An Overview," *Int. J. Data Warehous. Min.*, vol. 3, pp. 1–13, July 2007.

[73] M. T. Islam, M. A. Aowal, A. T. Minhaz, and K. Ashraf, "Abnormality Detection and Localization in Chest X-Rays using Deep Convolutional Neural Networks," 2017.

[74] Q. Guan, Y. Huang, Z. Zhong, Z. Zheng, L. Zheng, and Y. Yang, "Diagnose Like a Radiologist: Attention

Guided Convolutional Neural Network for Thorax Disease Classification," 2018.

[75] H. Liu, L. Wang, Y. Nan, F. Jin, Q. Wang, and J. Pu, "SDFN: Segmentation-Based Deep Fusion Network for Thoracic Disease Classification in Chest X-Ray Images," *Computerized Medical Imaging and Graphics*, vol. 75, pp. 66–73, July 2019.

[76] J. Cai, L. Lu, A. P. Harrison, X. Shi, P. Chen, and L. Yang, "Iterative Attention Mining for Weakly Supervised Thoracic Disease Pattern Localization in Chest X-Rays," in *Med. Image Comput. Comput. Assist. Interv. – MICCAI 2018* (A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, eds.), Lecture Notes in Computer Science, (Cham), pp. 589–598, Springer International Publishing, 2018.

[77] S. Guendel, F. C. Ghesu, S. Grbic, E. Gibson, B. Georgescu, A. Maier, and D. Comaniciu, "Multi-Task Learning for Chest X-Ray Abnormality Classification on Noisy Labels," May 2019.

[78] Z. Li, C. Wang, M. Han, Y. Xue, W. Wei, L.-J. Li, and L. Fei-Fei, "Thoracic Disease Identification and Localization With Limited Supervision," in *IEEECVF Conf. Comput. Vis. Pattern Recognit.*, (Salt Lake City, UT), pp. 8290–8299, IEEE, June 2018.

[79] C. Yan, J. Yao, R. Li, Z. Xu, and J. Huang, "Weakly Supervised Deep Learning for Thoracic Disease Classification and Localization on Chest X-Rays," in *Int. Conf. Bioinforma. Comput. Biol. Health Inform.*, (Washington DC USA), pp. 103–110, ACM, Aug. 2018.

[80] M. L. Zhang and Z. H. Zhou, "A Review on Multi-Label Learning Algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, pp. 1819–1837, Aug. 2014.

[81] K. Dembczyński, W. Waegeman, W. Cheng, and E. Hüllermeier, "On Label Dependence and Loss Minimization in Multi-Label Classification," *Mach Learn*, vol. 88, pp. 5–45, July 2012.

[82] H. Harvey and B. Glocker, "A Standardised Approach for Preparing Imaging Data for Machine Learning Tasks in Radiology," in *Artificial Intelligence in Medical Imaging: Opportunities, Applications and Risks* (E. R. Ranschaert, S. Morozov, and P. R. Algra, eds.), pp. 61–72, Cham: Springer International Publishing, 2019.

[83] W. Bi and J. T. Kwok, "Bayes-Optimal Hierarchical Multilabel Classification," *IEEE Trans. Knowl. Data Eng.*, vol. 27, pp. 2907–2918, Nov. 2015.

[84] H. Pourghassem and H. Ghassemian, "Content-Based Medical Image Classification Using a New Hierarchical Merging Scheme," *Computerized Medical Imaging and Graphics*, vol. 32, pp. 651–661, Dec. 2008.

[85] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Džeroski, "Hierarchical Annotation of Medical Images," *Pattern Recognition*, vol. 44, pp. 2436–2449, Oct. 2011.

[86] Y. Guo, Y. Liu, E. M. Bakker, Y. Guo, and M. S. Lew, "CNN-RNN: A Large-Scale Hierarchical Image Classification Framework," *Multimed Tools Appl*, vol. 77, pp. 10251–10271, Apr. 2018.

[87] K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, M. S. Gerber, and L. E. Barnes, "HDLTex: Hierarchical Deep Learning for Text Classification," in *16th IEEE Int. Conf. Mach. Learn. Appl. ICMLA*, (Cancun, Mexico), pp. 364–371, IEEE, Dec. 2017.

[88] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, (Honolulu, HI), pp. 6517–6525, IEEE, July 2017.

[89] D. Roy, P. Panda, and K. Roy, "Tree-Cnn: A Hierarchical Deep Convolutional Neural Network for Incremental Learning," *Neural Networks*, vol. 121, pp. 148–160, Jan. 2020.

[90] S. Van Eeden, J. Leipsic, S. F. Paul Man, and D. D. Sin, "The Relationship Between Lung Inflammation and Cardiovascular Disease," *Am J Respir Crit Care Med*, vol. 186, pp. 11–16, July 2012.

[91] N. Alaydie, C. K. Reddy, and F. Fotouhi, "Exploiting Label Dependency for Hierarchical Multi-Label Classification," in *Advances in Knowledge Discovery and Data Mining* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, P.-N. Tan, S. Chawla, C. K. Ho, and J. Bailey, eds.), vol. 7301, pp. 294–305, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[92] R. Aly, S. Remus, and C. Biemann, "Hierarchical Multi-Label Classification of Text With Capsule Networks," in *Proc. 57th Annu. Meet. Assoc. Comput. Linguist. Stud. Res. Workshop*, (Florence, Italy), pp. 323–330, Association for Computational Linguistics, 2019.

[93] W. Bi and J. T. Kwok, "Mandatory Leaf Node Prediction in Hierarchical Multilabel Classification," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, pp. 2275–2287, Dec. 2014.

[94] H. Chen, S. Miao, D. Xu, G. D. Hager, and A. P. Harrison, "Deep Hierarchical Multi-Label Classification

of Chest X-Ray Images," in *Proc. 2nd Int. Conf. Med. Imaging Deep Learn.*, pp. 109–120, PMLR, May 2019.

[95] H. Chen, S. Miao, D. Xu, G. D. Hager, and A. P. Harrison, "Deep Hiearchical Multi-Label Classification Applied to Chest X-Ray Abnormality Taxonomies," *Medical Image Analysis*, vol. 66, p. 101811, Dec. 2020.

[96] J. K. Gohagan, P. C. Prorok, R. B. Hayes, and B.-S. Kramer, "The Prostate, Lung, Colorectal and Ovarian (PLCO) Cancer Screening Trial of the National Cancer Institute: History, organization, and status," *Controlled Clinical Trials*, vol. 21, pp. 251S–272S, Dec. 2000.

[97] H. Q. Nguyen, K. Lam, L. T. Le, H. H. Pham, D. Q. Tran, D. B. Nguyen, D. D. Le, C. M. Pham, H. T. T. Tong, D. H. Dinh, C. D. Do, L. T. Doan, C. N. Nguyen, B. T. Nguyen, Q. V. Nguyen, A. D. Hoang, H. N. Phan, A. T. Nguyen, P. H. Ho, D. T. Ngo, N. T. Nguyen, N. T. Nguyen, M. Dao, and V. Vu, "VinDr-CXR: An Open Dataset of Chest X-Rays With Radiologist's Annotations," *Sci Data*, vol. 9, p. 429, July 2022.

[98] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognit. CVPR 2017*, 2017.

[99] J. P. Cohen, J. D. Viviano, P. Bertin, P. Morrison, P. Torabian, M. Guarrera, M. P. Lungren, A. Chaudhari, R. Brooks, M. Hashir, and H. Bertrand, "TorchXRayVision: A Library of Chest X-Ray Datasets and Models," in *Proc. 5th Int. Conf. Med. Imaging Deep Learn.*, pp. 231–249, PMLR, Dec. 2022.