DATA AND DOMAIN UNCERTAINTY IN MACHINE LEARNING

by

Artin Majdi

A Dissertation Submitted to the Faculty of the

Graduate Interdisciplinary Program
in Applied Mathematics

In Partial Fulfillment of the Requirements
For the Degree of

Doctor of Philosophy

In the Graduate College

The University of Arizona

May 28, 2023

Get the official approval page
from the Graduate College
before your final defense.

Statement by Author

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

Signed: _____

Dedication

For God.

Acknowledgments

First and foremost, thanks go to co-advisors Anna Mehmbuhr and Mbaqanga Uffda. Much of what I have learned in this project has been learned through them; their patience has been endless and their knowledge has been irreplaceable. Susan Kho and Vincenzo Mitti, as well, have given me valuable feedback as I have revised my dissertation this spring.

Summer 2009, fall 2009, and spring 2010 research assistantships were funded by the University of Arizona Department of Mathematics NSF VIGRE (Vertical InteGration in Research and Education) grant. The remaining semesters of my PhD years were funded by teaching assistantships. I thank my college-algebra, trigonometry, and calculus students for helping me to see mathematics from both sides of the classroom.

Table of Contents

List of Figures

List of Tables

Chapter 1

A Hierarchical multi-label Classification Technique for an Improved Diagnosis of Thoracic Disease in Chest Radiography

The accurate diagnosis of chest diseases based on chest radiographs is a difficult undertaking that can result in diagnostic errors that negatively impact patient outcomes. In this study, we propose a novel hierarchical multi-label classification method that employs a conditional loss function to enhance the identification of common thoracic diseases in chest radiographic images. Using a predefined disease taxonomy to account for interrelationships between diseases, the proposed method improves the classification performance of machine learning models. Our method can be seamlessly integrated into existing pre-trained models without requiring re-optimization, ensuring efficiency and extensive applicability. To determine the efficacy of the proposed method, experiments were conducted on a variety of publicly accessible datasets, including CheXpert, PadChest, and NIH Chest-Xray14. The results indicate that the proposed method substantially improves the precision and interpretability of machine learning models for thoracic disease on chest radiographs. This approach has the potential to improve patient outcomes by providing radiologists with an additional layer of decision support, thereby facilitating a more accurate and efficient diagnosis.

KEYWORDS: Chest radiography, hierarchical classification, disease taxonomy, multi-

label classification, conditional loss function, diagnostic error, machine learning, medical imaging

1.1 Introduction

Timely diagnosis and effective treatment of diseases hinge on the precise and efficient detection of anomalies in medical imaging. Deep learning techniques have made substantial progress in the medical imaging domain, exhibiting impressive success across various applications [**?**]. Nonetheless, conventional classification methods primarily designed for single-label predictions struggle with multi-label classification, which requires predicting multiple labels for each input sample.

Chest radiography (CXR) is a prevalent radiological examination for diagnosing lung and heart disorders, constituting a significant share of ordered imaging studies. Swift and accurate detection of different conditions, such as pneumothorax, is crucial for optimal patient care [**?**]. However, interpreting CXRs can be demanding due to the similarities between multiple diseases, which may result in misinterpretations even by seasoned radiologists [**?**]. Consequently, devising an accurate system to identify and localize common thoracic diseases can aid radiologists in minimizing diagnostic errors [**?**, **?**].

Convolutional neural networks (CNNs) exhibit potential for learning intricate relationships between image objects. However, their training necessitates vast quantities 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 capacity to execute complex tasks with minimal human intervention [**?**]. Progress in natural language processing (NLP) has enabled the gathering of extensive annotated datasets such as ChestX-ray8 [**?**], MIMIC-CXR [**?**], and CheXpert (Irvin et al., 2019b), allowing researchers to develop more efficient and robust supervised learning algorithms.

Regarding 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 [**?**]. 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 [**?**, **?**, **?**, **?**], leaving ample opportunities to investigate a unified deep learning framework that can efficiently detect a broad spectrum of common thoracic diseases.

This paper aims to tackle the challenges of multi-label classification within the realm of medical imaging by introducing a hierarchical framework that can be employed in a transfer learning approach 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 our proposed method for integrating label hierarchy into multi-label loss functions; Section 4 presents experimental results using the chest radiograph dataset; and Section 5 concludes the paper and outlines future research directions.

1.2 Related Work

The introduction of the ChestX-ray8 dataset and its associated model [**?**] 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 [**?**], attention mechanisms [**?**, **?**], and localization techniques [**?**, **?**, **?**, **?**]. 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 label, assuming independence between labels [**?**]. However, BR-based techniques do not account for label dependence, either conditional (instance-specific label dependence. In a given instance, the presence or absence of one label may impact another's.) or marginal (dataset-specific label dependence: certain labels may co-occur more frequently.) [**?**].

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 [**?**, **?**]. 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 [**?**]. Early methods used hierarchical and multi-label generalizations of traditional algorithms, such as nearest-neighbor or multi-layer perceptrons [**?**] and decision trees [**?**]. With the rise of deep learning, the adaptation of convolutional neural networks (CNN) for hierarchical classification has gained increasing attention [**?**, **?**, **?**, **?**].

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 contingent on the presence or absence of the parent labels in the hierarchy [**?**]. For example, if a radiologist is attempting to diagnose pneumonia in a patient, they may first look for evidence of lung consolidation (parent label) in the CXR, followed by specific patterns of lung consolidation suggesting pneumonia (child label). Consequently, it is possible to make more accurate diagnoses based on the data by considering the relationships 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 [**?**]. 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 [**?**], visual recognition [**?**], and genomic analysis [**?**]. A common technique [**?**] 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 [**?**]. These approaches help the classifier focus on the relevant data during initial training, improving prediction accuracy. It also allows the classifier to consider label hierarchies. 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. Additionally, they may not always perform satisfactorily, as it may not be possible to find a good set of parent-level labels that accurately capture the hierarchical relationships in the data. 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) [**?**, **?**].

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. The proposed loss function is based on the following hypothesis.

- The highest level of taxonomy contained the most general labels, whereas the lowest level contained the least.
- Each node contains a collection of granular child labels.

1.3 Methods

In this section, we present a comprehensive methodology to enhance multi-label classification performance using chest radiograph (CXR) data, which is applicable not only during the training phase, but also as a transfer learning approach during the testing phase. Our proposed strategy encompasses the formulation of a multi-label classification problem for CXR, the establishment of an evaluation protocol, and the incorporation of a loss measurement technique that leverages hierarchical label relationships. As a transfer learning approach, our method facilitates the adaptation and 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 in detecting abnormalities within CXR images.

1.3.1 Notations

Let us denote the following parameters:

- $C = \{c_k\}_{k=1}^{K}, c_k \in \{0, 1\}$: 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.
- $y_k^{(i)} \in \{0, 1\}$: true label for the k-th class(c_k) 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

- $t_k^{(i)} = \begin{cases} 1 & \text{if } p_k^{(i)} \geq \theta_k \\ 0 & \text{otherwise.} \end{cases}$: 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.

- $t_k^{(i)} = \begin{cases} 1 & \text{if } p_k^{(i)} \geq \theta_k \\ 0 & \text{otherwise.} \end{cases}$: 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, if 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 the i-th instance.

- $Y^{(i)} = \left\{y_k^{(i)}\right\}_{k=1}^{K}$: true label set, for instance i.

- $P^{(i)} = \left\{p_k^{(i)}\right\}_{k=1}^{K}$: predicted probability set, for instance i.

- $T^{(i)} = \left\{t_k^{(i)}\right\}_{k=1}^{K}$: 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 true labels.
- $\mathcal{L}\left(y_k^{(i)}, p_k^{(i)}\right)$: $\mathcal{L}$ is an arbitrary loss function (e.g., binary cross entropy) that takes the true label and predicted probability 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 in all cases 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.).
- $\mathcal{G} = \{\mathcal{C}, \mathcal{E}\}$: directed acyclic graph (DAG) $\mathcal{G}$ represents the taxonomy of thoracic diseases, where $\mathcal{C}$ is the set of disease classes and E is the set of directed edges representing parent-child relationships between these classes.
- $\Lambda(c_k) \subset \mathcal{C}$: set of parent classes of class c_k in DAG $\mathcal{G}$.
- $\mathcal{J}(c_k) \subset \mathcal{C}$: set of child classes of class c_k in DAG $\mathcal{G}$
- $\omega_k^{(i)}$: 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.
- $\widehat{p}_k^{(i)} = \omega_k^{(i)}\ p_k^{(i)}$: updated predicted probability for the k -th class.

1.3.2 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 $C = \{c_1, c_2, \ldots, c_K\}$, let us define a directed acyclic graph (DAG) $\mathcal{G} = \{C, \mathcal{E}\}$ representing the taxonomy of thoracic diseases, where E is the set of directed edges representing parent-child relationships between these classes. For each node $c_k \in C$, let $\Lambda(c_k) \in C : c_k \neq \Lambda(c_k)$ the parent node of class c_k and denote $\mathcal{J}(c_k) \subset C$ the set of child classes of class c_k in DAG $\mathcal{G}$. The root node does not represent an abnormality (ie, normal chest radiograph).

Let $\omega_k^{(i)}$ be a scalar weight assigned to the class c_k of instance i with respect to its parent class $\Lambda(c_k)$. 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 sample $X^{(i)}$ and class k.

1.3.3 Label Taxonomy and Hierarchy

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 categorized 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.

To develop a comprehensive label taxonomy for lung diseases, we integrated the taxonomies presented by Irvin [**?**] for the CheXpert dataset and Chen [**?**] for the PadChest and PLCO datasets. This unified taxonomical structure can be applied to various chest radiography datasets. In the following two sections, we propose two methods for incorporating taxonomy information to improve accuracy.

- In the first approach, we use taxonomy information to update the predicted probability of each class based on the predicted probability of its respective parent classes. This method can be easily applied unsupervised to existing, pre-trained models without the need for true labels.
- In our second approach, we propose a similar concept. 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.

1.3.4 Approach 1: Conditional Predicted Probability

A transfer learning-based approach that uses hyperparameters to update the predicted probability of a class based on the predicted probability of its parent class can be devised to further enhance the accuracy of classification by considering the interrelationship between different classes. In this approach, our aim is to calculate the conditional predicted probability for each class k and instance i, taking into account the predicted probabilities 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.

$$\widehat{p}_k^{(i)} = \frac{1}{1 + \exp\left(-\left(q_k^{(i)} + \alpha_{k,j} q_j^{(i)}\right)\right)} \tag{1.3.1}$$

where j is defined so that $c_j = \Lambda(c_k)$, and $\alpha_{k,j}$ is the hyperparameter controlling the influence of different parent class logits on child class logits. When $\alpha_{k,j} = 0$, there is no influence from the parent classes, and when $\alpha_{k,j} > 0$, it introduces a degree of dependency between the child and parent classes in terms of their predicted probabilities.

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 control 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.

Adaptive Computation for Real-World Applications The proposed transfer learning-based technique is adaptable to the user's computational capacity, making it suitable for real-world applications with varying computational resources. When sufficient computational resources are available, the method can be employed as a standalone loss function during the optimization process. On the other hand, when computational resources are limited, the 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 the calibration of the model output to more closely match the true label distribution.

1.3.5 Approach 2: Conditional Loss

Disadvantages of conditional predicted probability 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 the testing phase, it presents some challenges if we use it during the training phase of our classifier model. Among these disadvantages are the following.

- **Loss of interpretability:** Direct updates may obscure the effects of the optimization procedure, as the relationship between loss and predictions may become obscured.

- **Inconsistency with the optimization process:** Directly updating predicted probabilities may 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, thereby hindering generalization to unseen data.

Advantages of conditional loss function This approach offers several benefits in the context of multi-label classification tasks with hierarchical relationships.

- **Emphasis on error minimization:** Loss values represent the discrepancy between model predictions and ground truth labels. Incorporating parent class loss values into child class loss calculations focuses on minimizing errors across the hierarchy, thereby ensuring accurate predictions for both parent and child classes.
- **Enhanced gradient propagation:** Gradients are backpropagated through layers to update the model parameters during deep learning model training. Using parent class loss values in child class loss calculations strengthens the connection between parent and child classes in terms of gradient propagation, which could result in more efficient learning of hierarchical relationships and faster convergence during training.

- **Robustness to label noise:** Ground truth labels may contain inconsistencies or noise in real-world datasets. Incorporating parent class loss values into child class loss calculations promotes hierarchy consistency by penalizing deviations from expected parent-child relationships, thereby increasing the model's robustness to potential label noise within the dataset.

- **Improved interpretability:** Using loss values rather than predicted probabilities enables a more direct interpretation of the model's ability to capture hierarchical relationships between classes. High loss values for parent classes have a greater effect on the losses of their respective child classes, highlighting the need for improvement in certain areas to more accurately represent these relationships.

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:

$$l_k = -\left(y_k^{(i)} \log(p_k^{(i)}) + (1 - y_k^{(i)}) \log(1 - p_k^{(i)})\right) \tag{1.3.2}$$

$$\text{Loss}(\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} l_k \tag{1.3.3}$$

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 yield accurate predictions for multi-label classification tasks. However, class independence and equal importance between different classes cannot always be assumed (e.g., classification of thoracic diseases). In this study, we demonstrate how to incorporate the interdependence between different classes (i.e., thoracic diseases) into our loss measurement, consequently improving the overall classification accuracy. We introduce multiple approaches in which this can be achieved using either the parent class loss value or its predicted probability value.

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 hyperparameters. This term penalizes the loss for class k for each instance i in which the likelihood that its parent class exists in that instance is low. This can be represented mathematically by adding a hierarchical penalty term equal to the sum of the loss values of all parent classes of class k as follows.

$$\widehat{l}_k^{(i)} = l_k^{(i)} + \beta_k H(k|j) \tag{1.3.4}$$

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

$\Lambda(c_k)$ denote the set of parent classes for class k.

There are multiple ways to define the hierarchical penalty. For example, in one approach 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.

$$H(k|j) = \mathcal{L}\left(y_j^{(i)}, p_j^{(i)}\right) \tag{1.3.5}$$

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.

$$H\left(k|j\right) = \mathcal{L}\left(y_j^{(i)}, p_k^{(i)}\right) \tag{1.3.6}$$

In both Equations (**??**) and (**??**) the penalization term encourages the model to correctly predict the parent labels when predicting the child labels, ensuring that the predicted label set adheres to the hierarchical structure. In the aforementioned approaches, 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 and 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 issue, we can modify the loss measurements presented in Equations (**??**) and (**??**) 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 losses, 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.

$$\widehat{l}_k^{(i)} = l_k^{(i)} \; H\left(\; k \mid j \; \right) \tag{1.3.7}$$

where the hierarchical penalty term is defined as follows.

$$H(k|j) = \begin{cases} 1 & \text{if } \Lambda(c_k) = \varnothing \\ a_k \; l_j^{(i)} \; + \; \beta_k & \text{otherwise.} \end{cases} \tag{1.3.8}$$

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

The modified loss function in Equation (**??**) aims to ensure that predictions adhere to hierarchical relationships between labels by penalizing deviations from these established relationships. Adjusting the weighting parameters α_k and β_k, we can regulate the extent to which hierarchical information influences the learning process.

1.3.6 Updating Loss Values and Predicted Probabilities

In the previous section, we showed how a taxonomy-based loss function can be used to improve 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 our proposed taxonomy-based transfer learning approach 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 and predicted probabilities to incorporate the hierarchical relationships present in the taxonomy. We demonstrate how the interdependence between different classes, as represented by the hierarchical taxonomy, can be effectively captured through the adjustment of loss values and predicted probabilities. This ensures that the classifier's performance is enhanced while respecting the inherent structure of the disease taxonomy, ultimately leading to improved diagnosis and better patient outcomes.

To calculate the predicted probability based on the loss, we must first define a loss function that quantifies the difference between the predicted probability and the true label. Once the loss function has been defined, during a training phase of a classifier (e.g., DenseNet121), an optimization algorithm such as gradient descent can be 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 this section, we demonstrate how we can directly calculate the new predicted probabilities from their new loss values obtained in Equation (**??**). Let us assume that binary cross entropy is used for the choice of the loss function $\mathcal{L}(\cdot)$. Furthermore, let us denote $\widehat{q}_k^{(i)}, \widehat{p}_k^{(i)}$ as the updated values we are looking for showing the logit and predicted probability of class k and instance i after applying the proposed technique. As discussed before, to calculate the predicted probabilities, we need to pass the logits $\widehat{q}_k^{(i)}$ into a sigmoid function as shown below:

$$\widehat{p}_k^{(i)} = \text{sigmoid}\left(\widehat{q}_k^{(i)}\right) = \frac{1}{1 + \exp\left(-\widehat{q}_k^{(i)}\right)} \tag{1.3.9}$$

The sigmoid activation function maps any value to a number between zero and one. The sigmoid function is defined as follows. The gradient of the sigmoid function provides the direction in which the predicted probability must be updated.

$$\text{sigmoid}'\left(\widehat{q}_k^{(i)}\right) = \frac{\partial \text{sigmoid}}{\partial q} = \text{sigmoid}\left(\widehat{q}_k^{(i)}\right)\left(1 - \text{sigmoid}\left(\widehat{q}_k^{(i)}\right)\right) = \widehat{p}_k^{(i)}\left(1 - \widehat{p}_k^{(i)}\right) \tag{1.3.10}$$

The gradient of the loss 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.

$$\frac{\partial \mathcal{L}\left(\widehat{p}_k^{(i)},\ y_k^{(i)}\right)}{\partial \widehat{p}} = \frac{y_k^{(i)}}{\widehat{p}_k^{(i)}} - \frac{1 - y_k^{(i)}}{1 - \widehat{p}_k^{(i)}} \tag{1.3.11}$$

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 (**??**) and the derivative of the sigmoid function shown in Equation (**??**) to calculate the updated predicted probability.

$$\frac{\partial \mathcal{L}\left(p_k^{(i)},\ y_k^{(i)}\right)}{\partial p}\ \text{sigmoid}'\left(\widehat{q}_k^{(i)}\right) = \left(\frac{y_k^{(i)}}{\widehat{p}_k^{(i)}} - \frac{1 - y_k^{(i)}}{1 - \widehat{p}_k^{(i)}}\right)\widehat{p}_k^{(i)}\left(1 - \widehat{p}_k^{(i)}\right) = y_k^{(i)} - \widehat{p}_k^{(i)} \tag{1.3.12}$$

Hence, we can conclude the following.

$$\hat{p}_k^{(i)} = \begin{cases} -\frac{\partial \mathcal{L}\left(p_k^{(i)},\, y_k^{(i)}\right)}{\partial p}\ \text{sigmoid}'\left(\widehat{q}_k^{(i)}\right) + 1 & y = 1 \\ -\frac{\partial \mathcal{L}\left(p_k^{(i)},\, y_k^{(i)}\right)}{\partial p}\ \text{sigmoid}'\left(\widehat{q}_k^{(i)}\right) & \text{otherwise.} \end{cases} \tag{1.3.13}$$

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 (**??**) we will have the following:

$$\frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} = \frac{y_k^{(i)}}{\widehat{p}_k^{(i)}} - \frac{1 - y_k^{(i)}}{1 - \widehat{p}_k^{(i)}} = \frac{y_k^{(i)} - \widehat{p}_k^{(i)}}{\widehat{p}_k^{(i)}(1 - \widehat{p}_k^{(i)})} \tag{1.3.14}$$

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 (**??**) to look as follows.

$$\widehat{p}_k^{(i)} = \begin{cases} -\frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} \text{ sigmoid}'(q_k^{(i)}) + 1 & \text{if } \frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} \geq 0 \\ -\frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} \text{ sigmoid}'(q_k^{(i)}) & \text{otherwise.} \end{cases} \tag{1.3.15}$$

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

$$\widehat{p}_k^{(i)} = \begin{cases} \exp(-\widehat{l}_k^{(i)}) & \text{if } \frac{\partial \mathcal{L}(\widehat{p}_k^{(i)}, y_k^{(i)})}{\partial \widehat{p}} \geq 0 \\ 1 - \exp(-\widehat{l}_k^{(i)}) & \text{otherwise} \end{cases} \tag{1.3.16}$$

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

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

$$\exp\left(-\widehat{l}_k^{(i)}\right) = \exp\left(-l_k^{(i)} \left(\alpha_k \, l_j^{(i)} + \beta_k\right)\right) = \left(p_k^{(i)}\right)^{-\alpha_k \log\left(p_j^{(i)}\right) + \beta_k} \tag{1.3.17}$$

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

$$1 - \exp\left(-\widehat{l}_k^{(i)}\right) = 1 - \exp\left(-l_k^{(i)}\left(\alpha_k\ l_j^{(i)} + \beta_k\right)\right) = 1 - \left(1 - p_k^{(i)}\right)^{-\alpha_k \log\left(1-p_j^{(i)}\right)+\beta_k} \quad (1.3.18)$$

By substituting the Equations (**??**) and (**??**) into Equation (**??**) we will have the following.

$$\widehat{p}_k^{(i)} = \begin{cases} \left(p_k^{(i)}\right)^{-\alpha_k \log(p_j^{(i)})+\beta_k} & \text{if} \quad y_k^{(i)} = 1 \\ 1 - \left(1 - p_k^{(i)}\right)^{-\alpha_k \log\left(1-p_j^{(i)}\right)+\beta_k} & \text{otherwise} \end{cases} \quad (1.3.19)$$

1.3.7 Interpretability Enhancement

One of the key benefits of our proposed method 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's predictions, building trust in the model's outputs, and facilitating 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.

1.3.8 Experimental Setup

Datasets We utilized three diverse and publicly available datasets for the evaluation of our proposed hierarchical multi-label classification technique: CheXpert [**?**], PadChest [**?**], and VinDr-CXR [**?**]. These datasets contain a diverse range of chest radiographic images covering various thoracic diseases, providing a comprehensive evaluation of our method's effectiveness.

Dataset Description

- **CheXpert** [**?**] is a large-scale dataset containing 224,316 chest radiographs of 65,240 patients, labeled with 14 radiographic findings.
- **PadChest** [**?**] 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** [**?**] includes 112,120 chest radiographs of 30,805 patients labeled with 14 thoracic disease categories.

Preprocessing

The input 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, 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 Architecture and Training Details The pre-trained model provided by Cohen [**?**] was used as the base model. The model was fine-tuned using the DenseNet121 [**?**] architecture on a subset of CheXpert [**?**], NIH [**?**], PadChest [**?**] for 18 toracic 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 method, we utilized 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. In this investigation, we employed parallelization libraries, such as joblib and multiprocessing in Python, which enable the concurrent execution of multiple tasks while sharing available resources. These libraries facilitate the implementation of parallelism in our optimization process, ensuring seamless integration with the existing framework and offering a scalable and hardware-adaptable solution.

Optimum Threshold Determination

Determining the optimal threshold is a crucial aspect of evaluating the performance of our 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 opti-

mal 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 model's discriminatory ability 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 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 insights into the practical applicability of our approach in real-world settings.

Evaluation **Model Evaluation and Comparison**

After training, we evaluated the performance of our proposed method using the test set and compared it with other state-of-the-art methods for the multi-label classification of chest radiograph data. We use standard evaluation metrics, such as precision, recall, F1 score, and area under the receiver operating characteristic curve (AUROC), to assess the performance of our method. By incorporating the hierarchical relationship between different classes, our method aims to improve the performance of multi-label classification of chest radiograph data, leading to more accurate and efficient detection of thoracic abnormalities. This could improve the diagnosis and treatment of

these diseases in clinical practice. To demonstrate the effectiveness of our proposed hierarchical multi-label classification technique, we compare its performance to the baseline model trained without the proposed hierarchical scheme.

Evaluation Metrics

To assess the performance of the proposed techniques, we employ several evaluation metrics that capture different aspects of the classification problem. These metrics include precision, recall, F1 score, and the area under the receiver operating characteristic curve (AUC-ROC). By analyzing the performance of the method across these metrics, we can gain insights into the effectiveness of the technique in capturing inter-class relationships and improving classification accuracy. Additionally, the comparison of the proposed method with baseline approaches and other state-of-the-art techniques will provide a comprehensive understanding of the method's practical applicability and potential for real-world implementation. The following metrics were used to evaluate the effectiveness of the proposed method.

- **Accuracy**: Proportion of correctly classified samples to the total number of samples.
- **Precision**: Proportion of true positive predictions to the total number of positive predictions.
- **Recall**: Proportion of true positive predictions over the total number of actual positive instances.
- **F1-score**: The Harmonic mean of precision and recall, providing a balanced as-

sessment of the method's performance.

- **Area Under the Receiver Operating Characteristic Curve (AUROC)**: a summary measure of the true positive rate versus the false positive rate at different classification thresholds.

Dataset splits

The datasets were divided into four disjoint subsets:

- **Subset 1** used for model optimization consisting of training and validation. The training set was used to optimize the model parameters.
- **Subset2** used for applying the proposed label dependent technique comprised of training and testing. The training set was used for hyperparameter tuning. Test dataset was used to report the final findings.

1.4 Results

1.4.1 Distribution of Pathologies in Public Chest Radiograph Datasets

In this section, we present the distribution of different pathologies across three major public chest radiograph datasets: CheX, NIH, and PC. Table **??** compares the original and updated label sets for each dataset. The original counts represent the raw number of samples in the datasets, while the updated counts account for the absence of parent

Table 1.1. Comparison of the number of samples for different chest radiograph public datasets (CheX, NIH, and PC) per pathology, considering both original and updated label sets. The original counts represent the raw number of samples in the datasets, while the updated counts show the number of samples after updating the label set for parent pathologies when a dataset does not contain labels for that parent class. In the updated label set, a parent pathology is considered true if at least one of its child pathologies is present for that sample; otherwise, it is set to false.

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

pathology labels in some datasets by considering a parent pathology as true if at least one of its child pathologies is present for that sample.

In the original label sets, the total number of samples across the datasets were 20,543 for CheX, 28,868 for NIH, and 61,692 for PC. After updating the label sets, the total number of samples remained unchanged for CheX and NIH, while for PC, it increased to 61,692. The distribution of pathologies in the datasets varies, with some pathologies such as Atelectasis, Effusion, and Lung Opacity having a higher prevalence, while others like Hernia and Fibrosis showing a lower prevalence across the datasets.

Table 1: Details of the datasets included in this library. The number of images shows

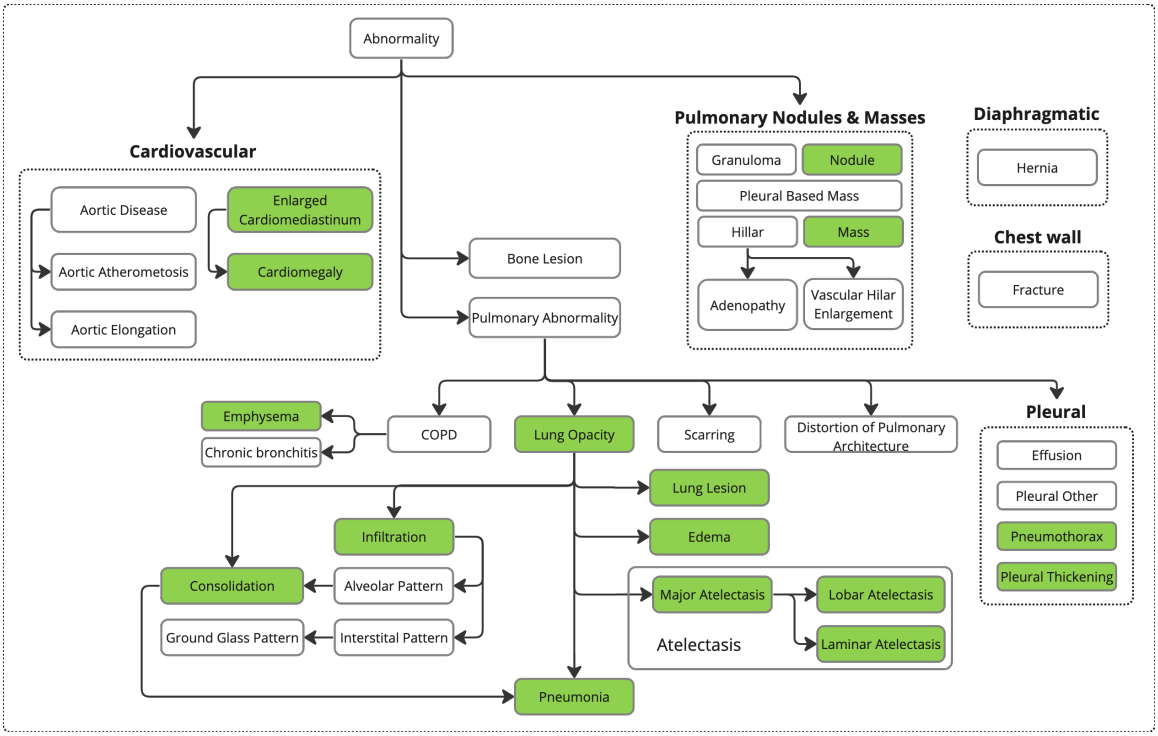

Figure 1.1. Taxonomy of lung pathologies on chest radiographs. This comprehensive classification system accumulated using taxonomy graphs in Irvin [**?**], and Chen [**?**] helps categorize various disease manifestations observed in public datasets, such as CheXpert, PadChest and PLCO and serves as a framework for understanding and analyzing chest radiograph abnormalities.

total images / usable frontal images. Usable frontal means images that are readable, have all the necessary metadata, and are in AP, PA, AP Supine, or AP Erect view.

Table 2: Labels available for each dataset, the total number of positive samples for each class across all datasets, and the total number of examples in each dataset, and the sum over each row in the right column. The COVID-19 datasets are excluded from this table because they have many unique pathologies.

1.4.2 Model Performance on Public Chest Radiograph Datasets

AUC In this section, we present the performance of the three methods—baseline, "logit", and "loss"; on the three public chest radiograph datasets (CheX, NIH, and PC) in terms of AUC metrics for various pathologies. The baseline represents the original model's performance, while "logit" and "loss" refer to the proposed modified logits and modified loss approaches, respectively. A single model was trained on all three datasets and evaluated on the test cases from each dataset **??**.

The results demonstrate varying performance across the pathologies and datasets. In general, the proposed "logit" and "loss" approaches show improved performance compared to the baseline, with several pathologies, such as Atelectasis, Consolidation, Edema, and Pneumonia, exhibiting significant improvements in AUC metrics. For instance, in the CheX dataset, the AUC for Atelectasis increased from 0.811 in the baseline to 0.960 and 1.000 in the "logit" and "loss" methods, respectively.

However, for some pathologies like Pneumothorax, Emphysema, and Pleural_Thickening,

Table 1.2. AUC performance of the three methods (baseline, "logit", and "loss") on the CheX, NIH, and PC chest radiograph datasets for various pathologies.

	CheX			NIH			PC		
pathologies\approach	**baseline**	**on_logit**	**on_loss**	**baseline**	**on_logit**	**on_loss**	**baseline**	**on_logit**	**on_loss**
Atelectasis	0.811	0.96	1	0.759	0.89	0.908	0.747	0.867	0.905
Consolidation	0.895	0.982	0.86	0.75	0.846	0.913	0.597	0.721	0.803
Infiltration				0.723	0.903	0.946	0.758	0.897	0.945
Pneumothorax	0.774	0.774	0.774	0.739	0.739	0.739	0.4	0.4	0.4
Edema	0.853	0.969	0.995	0.81	0.883	0.901	0.81	0.861	0.906
Emphysema				0.749	0.749	0.749	0.853	0.853	0.853
Fibrosis				0.775	0.775	0.775			
Effusion	0.872	0.872	0.872	0.866	0.866	0.866	0.847	0.847	0.847
Pneumonia	0.854	0.947	0.999	0.693	0.779	0.898	0.62	0.74	0.846
Pleural_Thickening				0.718	0.718	0.718	0.841	0.841	0.841
Cardiomegaly	0.86	0.911	0.998	0.859	0.986	0.96	0.776	0.97	0.911
Nodule				0.751	0.751	0.751	0.383	0.383	0.383
Mass				0.797	0.797	0.797	0.913	0.913	0.913
Hernia				0.999	0.999	0.999	0.806	0.806	0.806
Lung Lesion	0.788	0.93	1						
Fracture	0.736	0.736	0.736				0.742	0.742	0.742
Lung Opacity	0.804	0.804	0.804	0.742	0.742	0.742	0.782	0.782	0.782
Enlarged Cardiomediastinum	0.852	0.852	0.852	0.717	0.717	0.717	0.665	0.665	0.665

the performance remained consistent across all three approaches. In some cases, such as Mass and Nodule in the PC dataset, the performance was notably lower than in the other datasets.

F1-score In this section, we discuss the F1-score performance of the three methods—baseline, "logit", and "loss"—on the three public chest radiograph datasets (CheX, NIH, and PC) for various pathologies **??**. The baseline represents the original model's performance, while "logit" and "loss" refer to the proposed modified logits and loss approaches, respectively. As before, a single model was trained on all three datasets and evaluated on the test cases from each dataset.

The F1-scores reveal that the proposed "logit" and "loss" approaches generally show improved performance compared to the baseline method. For example, in the CheX

Table 1.3. F1-score performance of the three methods (baseline, “logit”, and “loss”) on the CheX, NIH, and PC chest radiograph datasets for various pathologies.

	CheX			NIH			PC		
pathologies\approach	**baseline**	**on_logit**	**on_loss**	**baseline**	**on_logit**	**on_loss**	**baseline**	**on_logit**	**on_loss**
Atelectasis	0.201	0.353	0.927	0.007	0.123	0.007	0.079	0.4	0.081
Consolidation	0.207	0.784	0.188	0	0.048	0	0	0.069	0
Infiltration				0.058	0.325	0.059	0.204	0.575	0.214
Pneumothorax	0	0	0	0	0	0	0	0	0
Edema	0.352	0.733	0.829	0	0	0	0.107	0.241	0.115
Emphysema				0	0	0	0	0	0
Fibrosis				0	0	0			
Effusion	0.328	0.328	0.328	0.27	0.27	0.27	0.35	0.35	0.35
Pneumonia	0.156	0.759	0.623	0.056	0.078	0.078	0.192	0.26	0.224
Pleural_Thickening				0	0	0	0	0	0
Cardiomegaly	0.432	0.46	0.889	0.116	0.333	0.125	0.305	0.519	0.396
Nodule				0.014	0.014	0.014	0	0	0
Mass				0.278	0.278	0.278	0	0	0
Hernia				0	0	0	0.267	0.267	0.267
Lung Lesion	0.126	0.602	0.69						
Fracture	0.124	0.124	0.124				0	0	0
Lung Opacity	0.345	0.345	0.345	0.446	0.446	0.446	0.537	0.537	0.537
Enlarged Cardiomediastinum	0.425	0.425	0.425	0	0	0	0.025	0.025	0.025

dataset, the F1-score for Atelectasis increased from 0.201 in the baseline to 0.353 and 0.927 in the “logit” and “loss” methods, respectively. Similarly, the F1-score for Edema increased from 0.352 in the baseline to 0.733 and 0.829 in the “logit” and “loss” methods, respectively.

However, for some pathologies such as Pneumothorax, Emphysema, and Pleural_Thickening, the F1-scores remained consistently low across all three approaches. This result indicates that there is room for further improvement in these areas.

The accuracy of the three methods (baseline, “logit”, and “loss”) was evaluated for the CheX, NIH, and PC datasets for different pathologies, as presented in Table **??**. Comparing the baseline method to the “logit” and “loss” methods, we observed improvements in accuracy across most pathologies in all three datasets. For instance,

Table 1.4. Accuracy performance of the three methods (baseline, "logit", and "loss") on the CheX, NIH, and PC chest radiograph datasets for various pathologies.

	CheX			NIH			PC		
pathologies\approach	**baseline**	**on_logit**	**on_loss**	**baseline**	**on_logit**	**on_loss**	**baseline**	**on_logit**	**on_loss**
Atelectasis	0.593	0.796	0.992	0.89	0.895	0.89	0.905	0.885	0.907
Consolidation	0.81	0.984	0.789	0.972	0.971	0.972	0.952	0.926	0.955
Infiltration				0.88	0.887	0.882	0.765	0.819	0.779
Pneumothorax	0.983	0.983	0.983	0.973	0.973	0.973	0.995	0.995	0.995
Edema	0.768	0.948	0.973	0.972	0.971	0.972	0.932	0.914	0.937
Emphysema				0.98	0.98	0.98	0.988	0.988	0.988
Fibrosis				0.994	0.994	0.994			
Effusion	0.678	0.678	0.678	0.925	0.925	0.925	0.858	0.858	0.858
Pneumonia	0.938	0.994	0.992	0.975	0.957	0.983	0.862	0.806	0.887
Pleural_Thickening				0.982	0.982	0.982	0.989	0.989	0.989
Cardiomegaly	0.796	0.788	0.978	0.978	0.982	0.979	0.888	0.929	0.925
Nodule				0.948	0.948	0.948	0.959	0.959	0.959
Mass				0.933	0.933	0.933	0.985	0.985	0.985
Hernia				1	1	1	0.985	0.985	0.985
Lung Lesion	0.874	0.983	0.991						
Fracture	0.943	0.943	0.943				0.975	0.975	0.975
Lung Opacity	0.564	0.564	0.564	0.74	0.74	0.74	0.72	0.72	0.72
Enlarged Cardiomediastinum	0.838	0.838	0.838	0.975	0.975	0.975	0.895	0.895	0.895

in the CheX dataset, the accuracy for Atelectasis increased from 0.593 in the baseline method to 0.796 and 0.992 in the "logit" and "loss" methods, respectively. Similarly, in the NIH dataset, the accuracy for Edema improved from 0.972 in the baseline method to 0.971 and 0.972 for the "logit" and "loss" methods, respectively.

In some cases, the "logit" and "loss" methods achieved comparable performance, while in others, one method outperformed the other. For example, in the PC dataset, the accuracy for Pneumonia improved from 0.862 in the baseline to 0.806 and 0.887 in the "logit" and "loss" methods, respectively, with the "loss" method yielding higher accuracy. These results demonstrate the effectiveness of the proposed modifications in improving the performance of the models across various chest radiograph pathologies.

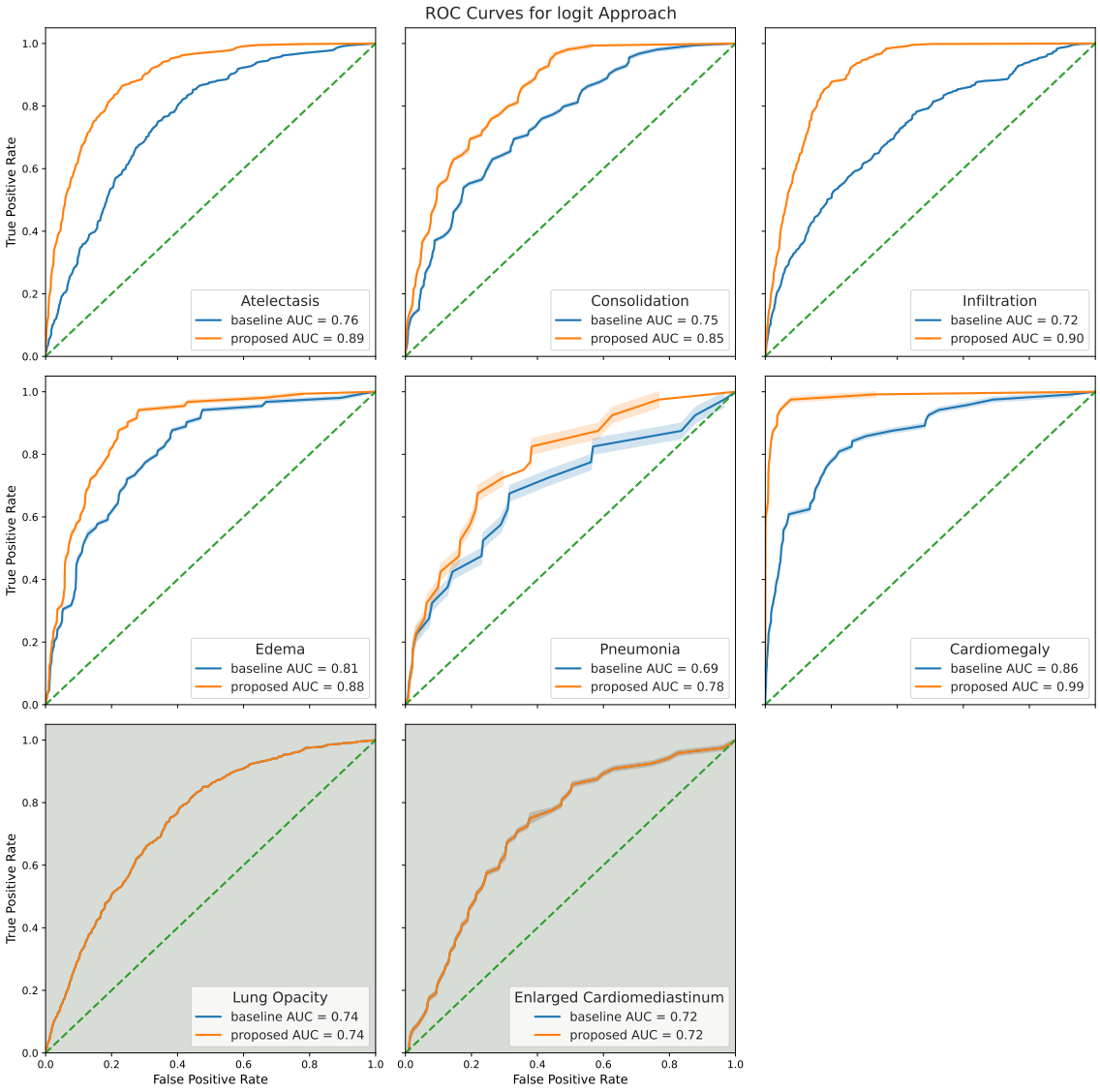

Figure 1.2. Comparative analysis of the ROC curves for eight thoracic pathologies using the baseline and the logit based proposed technique. Each subplot illustrates the overlaid ROC curves for both techniques pertaining to a specific pathology. The subplots highlighted with a darker background, represent parent class diseases.

Figure **??** illustrates a comparison between the performance of the baseline technique and the proposed logit based method (Approach 1 discussed in the Method section) in detecting eight thoracic pathologies. These eight pathologies includes the pathologies with child classes and their respective child classes as was shown in Figure **??**. The individual subplots exhibit overlaid receiver operating characteristic (ROC) curves, each of which corresponds to a specific pathology. The present analysis employed a model that was derived through the application of the test dataset from the NIH dataset to a pretrained model. The latter had undergone training on various publicly available thoracic datasets, with the aim of enabling the identification of 18 different pathologies pertaining to the thorax. The last two subplots showcased with a darker background showcases the roc curves for parent classes diseases. As these parent class diseases were not influenced by the proposed technique, their ROC curves and corresponding Area Under the Curve (AUC) values remain consistent with the baseline technique. The ROC curve and its corresponding AUC for the six child classes demonstrate a significant improvement for the proposed technique in comparison to the baseline technique.

Figure **??** illustrates a comparison between the performance of the baseline technique and the proposed loss based method (Approach 2 discussed in the Method section) in detecting eight thoracic pathologies. These eight pathologies includes the pathologies with child classes and their respective child classes as was shown in Figure **??**. The individual subplots exhibit overlaid ROC curves, each of which corresponds to a specific pathology. The present analysis employed a model that was derived through the application of the test dataset from the NIH dataset to a pretrained model. The

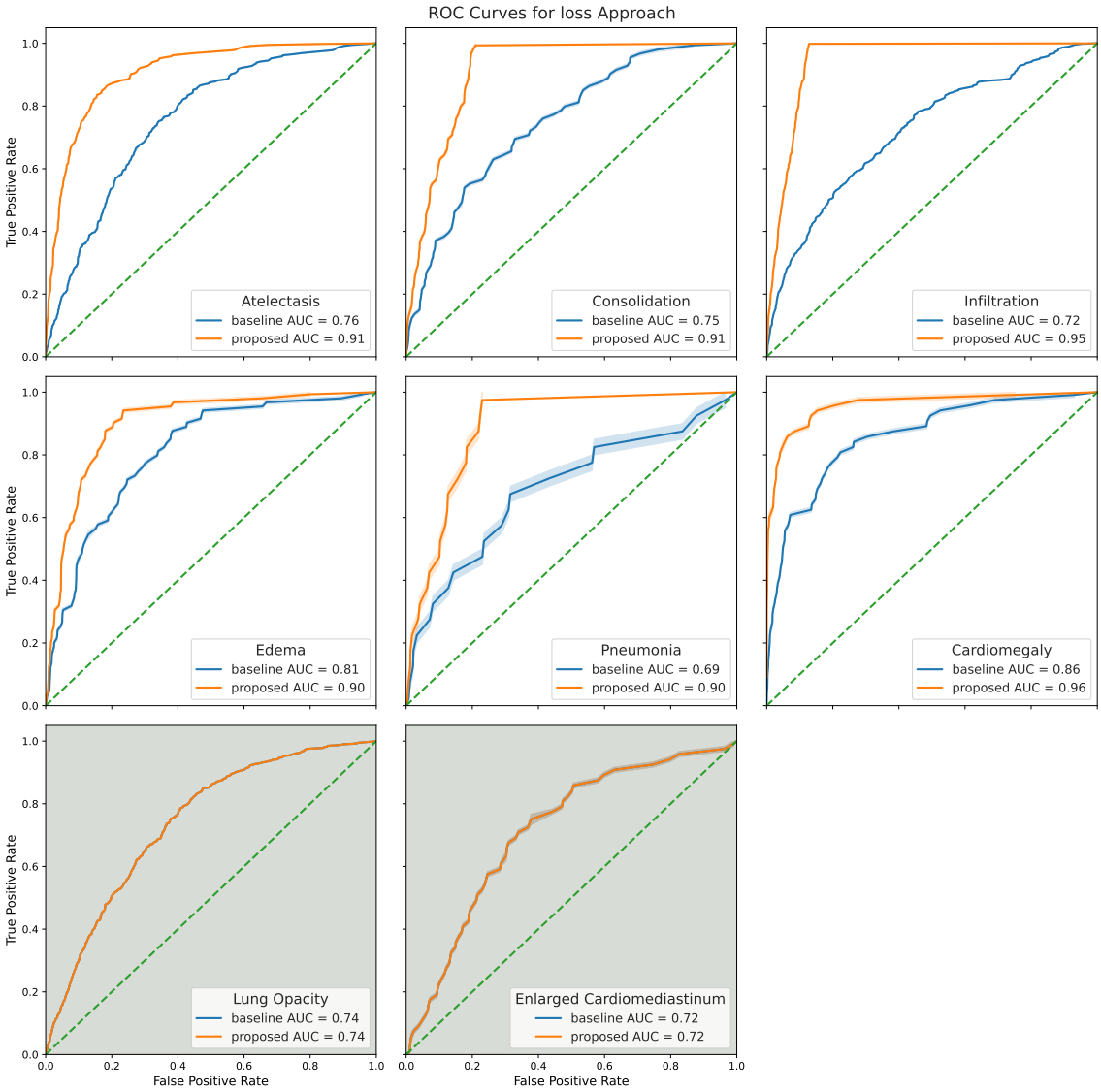

Figure 1.3. Comparative analysis of the ROC curves for eight thoracic pathologies using the baseline and the loss based proposed technique. Each subplot illustrates the overlaid ROC curves for both techniques pertaining to a specific pathology. The subplots highlighted with a darker background, represent parent class diseases.

latter had undergone training on various publicly available thoracic datasets, with the aim of enabling the identification of 18 different pathologies pertaining to the thorax. The last two subplots showcased with a darker background showcases the roc curves for parent classes diseases. As these parent class diseases were not influenced by the proposed technique, their ROC curves and corresponding AUC values remain consistent with the baseline technique. The ROC curve and its corresponding AUC for the six child classes demonstrate a significant improvement for the proposed technique in comparison to the baseline technique.

1.5 Discussion and Conclusion

1.5.1 Potential Applications

The proposed hierarchical multi-label classification method has several potential applications within the medical imaging domain:

1. **Assisting radiologists**: The method can be used as a decision support tool for radiologists, providing them with additional insights and reducing the likelihood of diagnostic errors.
2. **Automated triage**: The method can be used in emergency departments to prioritize patients according to the severity of their thoracic diseases, ensuring that the most urgent cases are attended to promptly.

3. **Population health management**: The method could be used for the population-level screening of thoracic diseases, helping healthcare organizations identify and manage outbreaks of diseases such as pneumonia or tuberculosis.

Appendices

Acknowledgements

Chapter 2

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

2.1 Introduction

Supervised learning techniques require a large amount of labeled data to train models to classify new data [**?**, **?**]. Traditionally, data labeling has been assigned to experts in the domain or well-trained annotators [**?**]. Although this method produces high-quality labels, it is inefficient and costly [**?**, **?**]. 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 [**?**, **?**] and natural language processing [**?**]. 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 [**?**]. Several studies have demonstrated the efficacy of repeated labeling [**?**, **?**]. 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 [**?**] discovered that Raykar's [**?**] technique outperformed other aggregation techniques. However, according to another study [**?**], the traditional Dawid-Skene (DS) model [**?**] 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 [**?**]. 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 [**?**]. 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 [**?**]. 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 [**?**]. 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 and their accuracy with respect to other annotators. 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 experimental results show that the proposed technique generates a weight that closely follows the annotator's degree of reliability. Furthermore, 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 a 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.

The remainder of this paper is organized as follows. Section 2 examines related work involving label aggregation algorithms. In Section 3, we provide a detailed explanation of our proposed technique. Section 4 presents the experiments and findings. Finally, Section 5 summarizes the findings and identifies the main directions for future research.

2.2 Related Work

Numerous label aggregation algorithms have been developed to capture the complexity of crowdsourced labeling systems, including techniques based on annotator reliability [**?**,**?**], confusion matrices [**?**,**?**], intentions [**?**,**?**], biases [**?**,**?**,**?**], and correlations [**?**]. However, because crowdsourced labeling is inherently dynamic and uncertain, developing a technique that can work in most situations is extremely challenging. Many techniques [**?**,**?**,**?**,**?**,**?**] utilize the Dawid and Skene (DS) generative model [**?**]. Ghosh [**?**] extended the DS model by using singular value decomposition (SVD) to calculate the reliability of the annotator. Similarly to Ghosh [**?**], Dalvi [**?**] 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 [**?**] described an iterative strategy for binary labeling based on a one-coin model [**?**]. Karger [**?**] 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 [**?**] proposed simulta-

neous 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 [**?**,**?**,**?**,**?**,**?**,**?**,**?**,**?**]. 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 [**?**]. 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 [**?**]. 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 [**?**]. 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 im-

prove the annotation [**?**]. Recently, Sheng [**?**] 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 [**?**] improved Sheng's approach by assigning different weights to annotators for each instance. This weighting method combines the specific quality $s_\alpha^{(i)}$ for the annotator α and instance i and the overall quality τ_α across all instances. Inspired by Li's technique [**?**], 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 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 [**?**] 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 [**?**] 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.

2.3 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 while having low accuracy), we extend the proposed technique by penalizing the annotators for instances in which they disagree with the aggregated label obtained using the 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 demonstrated in the experimental sections, the proposed crowd-certain method is not only comparable to other techniques in terms of accuracy 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.

2.3.1 Glossary of Symbols

Let us denote the following parameters:

N: Number of instances.

M: Number of annotators.

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

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

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

$\pi_{\alpha,k}$: Reliability score to generate sample labels for annotator α for class k. For example, it 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.

$Y^{(i)} = \left\{y_1^{(i)}, y_2^{(i)}, \ \ldots, y_K^{(i)}\right\}$: 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 $Y^{(i)} = \{1, 1, 0\}$.

$Z_\alpha^{(i)} = \left\{z_{a,1}^{(i)}, \ z_{a,2}^{(i)}, \ \ldots, \ z_{a,K}^{(i)}\right\}$: 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 will be 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 reliability score for that class $\Pi_{\alpha,k}$ is greater than $\rho^{(i)}$, the true label $y_k^{(i)}$ will be assigned; otherwise, an incorrect label $1 - y_k^{(i)}$ will be assigned.

$\Pi_\alpha = \left\{\pi_{\alpha,1}, \ \pi_{\alpha,2}, \ \ldots, \ \pi_{\alpha,K}\right\}$: set of K reliability scores 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)

$P_\alpha^{(i,g)} = \left\{p_{\alpha,k}^{(i,g)}\right\}_{k=1}^{K}$: 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).

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

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

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

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

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

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

2.3.2 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:

$$E = \sum_{i=1}^{N}\sum_{a=1}^{M}\left(\sum_{k=1}^{K}\delta\left(y_k^{(i)}, z_{\alpha,k}^{(i)}\right)\right) \tag{2.3.1}$$

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 misclas-

sification. 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 the entire dataset, capturing the performance of the aggregation model on 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, 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 as follows:

$$R(h) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(Y^{(i)}, h\left(\left\{Z_{\alpha}^{(i)}\right\}_{\alpha=1}^{M}\right)\right) \tag{2.3.2}$$

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 [**?**], as shown below:

$$\widehat{h} = \underset{h}{\text{argmin}}, R(h) \tag{2.3.3}$$

2.3.3 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 $M \times K$ reliability values 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 reliability values 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 $z_{\alpha,k}^{(i)}$ for each annotator α, instance i and class k are as follows:

$$z_{\alpha,k}^{(i)} = \begin{cases} y_k^{(i)} & \text{if } \rho^{(i)} \leq \pi_{\alpha,k}, \\ 1 - y_k^{(i)} & \text{if } \rho^{(i)} > \pi_{\alpha,k}, \end{cases} \quad \forall i, a, k \tag{2.3.4}$$

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

2.3.4 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 $X^{(i)}$ [**?**]. 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 is 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 (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 $\left\{p_{\alpha,k}^{(i,g)}\right\}_{k=1}^{K}$ representing the probability that class k is present in the sample. Consequently, we obtain a collection of G predicted probability sets $\left\{\left\{p_{\alpha,k}^{(i,g)}\right\}_{k=1}^{K}\right\}_{g=1}^{G}$ for each annotator α and instance i. The set $\left\{p_{\alpha,k}^{(i,g)}\right\}_{g=1}^{G}$ contains the predicted probabilities for class k, annotator α, and instance i. Disagreements between predicted probabilities $\left\{p_{\alpha,k}^{(i,g)}\right\}_{g=1}^{G}$ 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 $t_{\alpha,k}^{(i,g)}$ as the predicted label obtained by binarizing the predicted probabilities $p_{\alpha,k}^{(i,g)}$ 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 $u_{\alpha,k}^{(i)}$ 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 denotes 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:

$$u_{\alpha,k}^{(i)} = H\left(\left\{p_{\alpha,k}^{(i,g)}\right\}_{g=1}^{G}\right) = -\sum_{g} p_{\alpha,k}^{(i,g)} \log\left(p_{\alpha,k}^{(i,g)}\right) \tag{2.3.5}$$

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 $\{t_{\alpha,k}^{(i,g)}\}_{g=1}^{G}$ with mean value μ, the standard deviation is defined as

$$u_{\alpha,k}^{(i)} = \mathrm{SD}\left(\left\{t_{\alpha,k}^{(i,g)}\right\}_{g=1}^{G}\right) = \sqrt{\frac{1}{G-1}\sum_{g=1}^{G}\left(t_{\alpha,k}^{(i,g)} - \mu\right)^2}, \quad \mu = \frac{1}{G}\sum_{g=1}^{G} t_{\alpha,k}^{(i,g)} \tag{2.3.6}$$

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:

$$\begin{aligned} P\left(Q_L^k \leq p_{\alpha,k}^{(i,g)} \leq Q_U^k\right) &= \gamma \\ u_{\alpha,k}^{(i)} &= Q_L^k - Q_U^k \end{aligned} \tag{2.3.7}$$

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 $P_{\alpha,k}^{(i)} = \mathrm{sorted}\left(\left\{p_{\alpha,k}^{(i,g)}\right\}_{g=1}^{G}\right), \quad \forall \alpha, k, i.$
2. Calculate the lower and upper quantile indices based on the chosen confidence level γ. The lower quantile index is $L = \mathrm{ceil}\left(\frac{G}{2}(1-\gamma)\right)$, 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 $P^{(i)}_{\alpha,k}$. These values are the lower and upper quantiles Q^k_L and Q^k_U.

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

Monte Carlo Dropout The Monte Carlo dropout [**?**] 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 [**?**]. 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 [**?**] 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 Committee-based method [**?**] 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 in-

clude bagging and boosting ensemble methods and models, such as random forests.

$$u_{\alpha,k}^{(i)} = \mathrm{VarCommittee}\left(P_{\alpha,k}^{(i)}\right) = \frac{1}{G}\sum_{g=1}^{G}\left(p_{\alpha,k}^{(i,g)} - \mu\right)^2, \quad \mu = \frac{1}{G-1}\sum_{g=1}^{G} p_{\alpha,k}^{(i,g)} \tag{2.3.8}$$

Conformal Prediction: Conformal prediction [**?**] 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 $\eta_{\alpha,k}^{(i)}$. 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 $\zeta_k^g = \mathrm{score\_function}\left(p_{\alpha,k}^{(i,g)}, y_k^{(i)}\right)$

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

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 (**??**) as our uncertainty measurement due to its simplicity. However, other measures could also be employed as suitable alternatives.

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

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

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

$$c_{\alpha,k}^{(i)} = 1 - u_{\alpha,k}^{(i)} \ , \ \forall i, \alpha, k \tag{2.3.9}$$

2. In a second approach (shown in Equation (**??**)), we penalize annotators for instances in which their predicted label $\eta_{\alpha,k}^{(i)}$ (explained in the Glossary of Symbols section) does not match the MV of all annotator labels $\underset{\alpha}{\mathrm{MV}}\left(z_{\alpha,\,k}^{(i)}\right)$. As previously discussed, instead of directly working with the annotator's labels $z_{\alpha,k}^{(i)}$, we use the

predicted labels obtained from the ensemble of classifiers $\eta_{\alpha,k}^{(i)}$. 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.

$$c_{\alpha,k}^{(i)} = \begin{cases} 1 - u_{\alpha,k}^{(i)} & \text{if } \eta_{\alpha,k}^{(i)} = \mathrm{MV}_{\alpha}(\eta_{\alpha,k}^{(i)}) \\ 0 & \text{otherwise} \end{cases} \tag{2.3.10}$$

Weight Measurement Furthermore, we define the annotators' weights $\omega_{\alpha,k}$ for each class k as the mean of their respective consistency scores $c_{\alpha,k}^{(i)}$ over all instances. The weights are normalized between 0 and 1 as follows:

$$\omega_{\alpha,k} = \frac{\psi_{\alpha,k}}{\sum_{\alpha=1}^{M} \psi_{\alpha,k}} \quad \text{where} \quad \psi_{\alpha,k} = \frac{1}{N} \sum_{i=1}^{N} c_{\alpha,k}^{(i)} \tag{2.3.11}$$

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

$$v_k^{(i)} = \begin{cases} 1 & \text{if } \left(\sum_{\alpha=1}^{M} \omega_{\alpha,k}\, \eta_{\alpha,k}^{(i)} \right) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, k \tag{2.3.12}$$

Confidence Score Calculation In previous section we showed how to calculate the aggregated label $v_k^{(i)}$ (shown in Equation (**??**)). Define $F_k^{(i)}$ as the confidence score for instance i and class k. In this section, we calculate two confidence scores $F_k^{(i)}$, based on how many different annotators agree on the reported label $v_k^{(i)}$, which will be re-

ported alongside the aggregated label $v_k^{(i)}$. 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 [**?**] and Tao [**?**] to incorporate our calculated weight $\omega_{\alpha,k}$ shown in Equation (**??**) for each worker α.

1. **uwMV-Freq:** In this approach, the confidence score $F^{(i)}$ is defined as the weighted sum of labels belonging to annotators whose label is the same as the final aggregated label. It is calculated as

$$F_k^{(i)} = \sum\nolimits_{\alpha=1}^{M} \omega_{\alpha,k}\ \delta\left(\eta_{\alpha,k}^{(i)}\ \ ,\ \ \ v_k^{(i)}\right) \tag{2.3.13}$$

   where δ is the Kronecker delta function.

2. **uwMV-Beta:** In this approach, the CDF of the beta distribution at the decision threshold of 0.5 is used to calculate a confidence score $F^{(i)}$. To calculate the two shape parameters of the beta distributions $\alpha^{(i)}$ and $\beta^{(i)}$, we use a weighted sum of all correct and incorrect aggregated labels, respectively:

$$\begin{aligned} l_k^{(i)} &= 1 + \sum_{\alpha=1}^{M} \omega_{\alpha,k}\ \delta\left(\eta_{\alpha,k}^{(i)}, v_k^{(i)}\right) \\ u_k^{(i)} &= 1 + \sum_{\alpha=1}^{M} \omega_{\alpha,k}\ \delta\left(\eta_{\alpha,k}^{(i)}, 1 - v_k^{(i)}\right) \end{aligned} \tag{2.3.14}$$

$$F_k^{(i)} = I_{0.5}\left(l_k^{(i)}, u_k^{(i)}\right) = \sum_{t=\lfloor l_k^{(i)} \rfloor}^{T-1} \frac{(T-1)!}{t!(T-1-t)!} 0.5^{T-1} \tag{2.3.15}$$

where $T = \left\lfloor l_k^{(i)} + u_k^{(i)} \right\rfloor$ and $\lfloor \cdot \rfloor$ is the floor function.

2.4 Results

In this section, we assess the efficacy of our proposed strategy. To evaluate our proposed technique, we conducted a series of experiments comparing the proposed technique with existing state-of-the-art techniques such as MV, Tao [**?**], and Sheng [**?**], as well as with other crowdsourcing methodologies reported in the crowd-kit package [**?**] including Gold Majority Voting, MMSR [**?**], Wawa, Zero-Based Skill, GLAD [**?**], and Dawid Skene [**?**].

2.4.1 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 **??** provides an overview of the datasets used. All datasets are obtained from the University of California, Irvine (UCI) repository [**?**].

- 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%).

Table 2.1. 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 **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 nine attributes corresponding to the tic-tac-toe squares: x, o, and b (blank).

- 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. 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. In this study, we generated multiple fictitious label sets for each dataset to simulate the crowdsourcing concept of collecting several crowd labels for each instance. This was achieved by selecting random samples in the datasets using a uniform distribution and altering 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 varying levels of accuracy, we aim to evaluate the performance of our proposed method under different conditions, such as varying annotator expertise and reliability. This process allowed 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.

2.4.2 Benchmarks:

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

- **Golden majority voting** estimates the probability that each annotator possesses the correct label and then calculates the probability of each label per occurrence based on the weight assigned to each label. Assume that 10,000 jobs are performed by 3,000 annotators as well as 100 instances of ground truth. First, the percentage of correct labels for each annotator is calculated. The remaining labels were then estimated based on the weights.

- **Wawa** (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** employs a weighted majority vote (WMV). After processing a collection of instances, it re-evaluated 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.
- Descriptions of the other techniques can be found in their respective references.

2.4.3 Weight Measurement:

After generating the multi-label sets, we employed both the proposed and state-of-the-art approaches to obtain the aggregated labels. We experimented with two approaches for classifier selection, as explained in Section 3.4.1. We found no significant differences in the overall outcomes and thus chose the second approach, which utilized the Random Forest classification technique, to save processing time and reduce the number of required Python package dependencies. Ten Random Forests, each with four trees and a maximum depth of four, were trained in different random states for each annotator α, as detailed in Section 3.

Annotators' reliability vsėstimated weight $\omega_{\alpha,k}$

Figure **??** depicts the relationship between the randomly assigned annotators' relia-

bility value ($\pi_{\alpha,k}$) and their corresponding estimated weights, $\omega_{\alpha,k}$. In the case of Tao's method, the figure displays the average weights across all instances. As seen, when the reliability of an annotator surpasses a specific threshold, the weight measured by Tao's technique plateaus, whereas the proposed method exhibits a considerably stronger correlation. Individual data points represent the actual measured weights, and the curve represents the regression line.

2.4.4 Confidence-score

The results present a box plot of the average accuracy for different numbers of annotators, ranging from three to ten. Figure **??** illustrates the average accuracy of the crowd-certain technique with penalization for both the "freq" and "Beta" confidence measurement approaches. A noticeable difference in the accuracy was observed. However, statistical analysis did not reveal a significant difference between these approaches.

Figure **??** displays the average accuracy using the "freq" confidence measurement strategy for the proposed crowd-certain technique with and without penalization. The penalization method occurs by penalizing annotators for inaccurate labeling before measuring their weights, as demonstrated in Equation (**??**). The penalized version of the proposed technique shows an improvement in average accuracy and a reduction in variance.

Figure **??** shows the average accuracy distribution ("freq" strategy) of the proposed penalized versus Tao and Sheng for a different number of annotators using the kernel

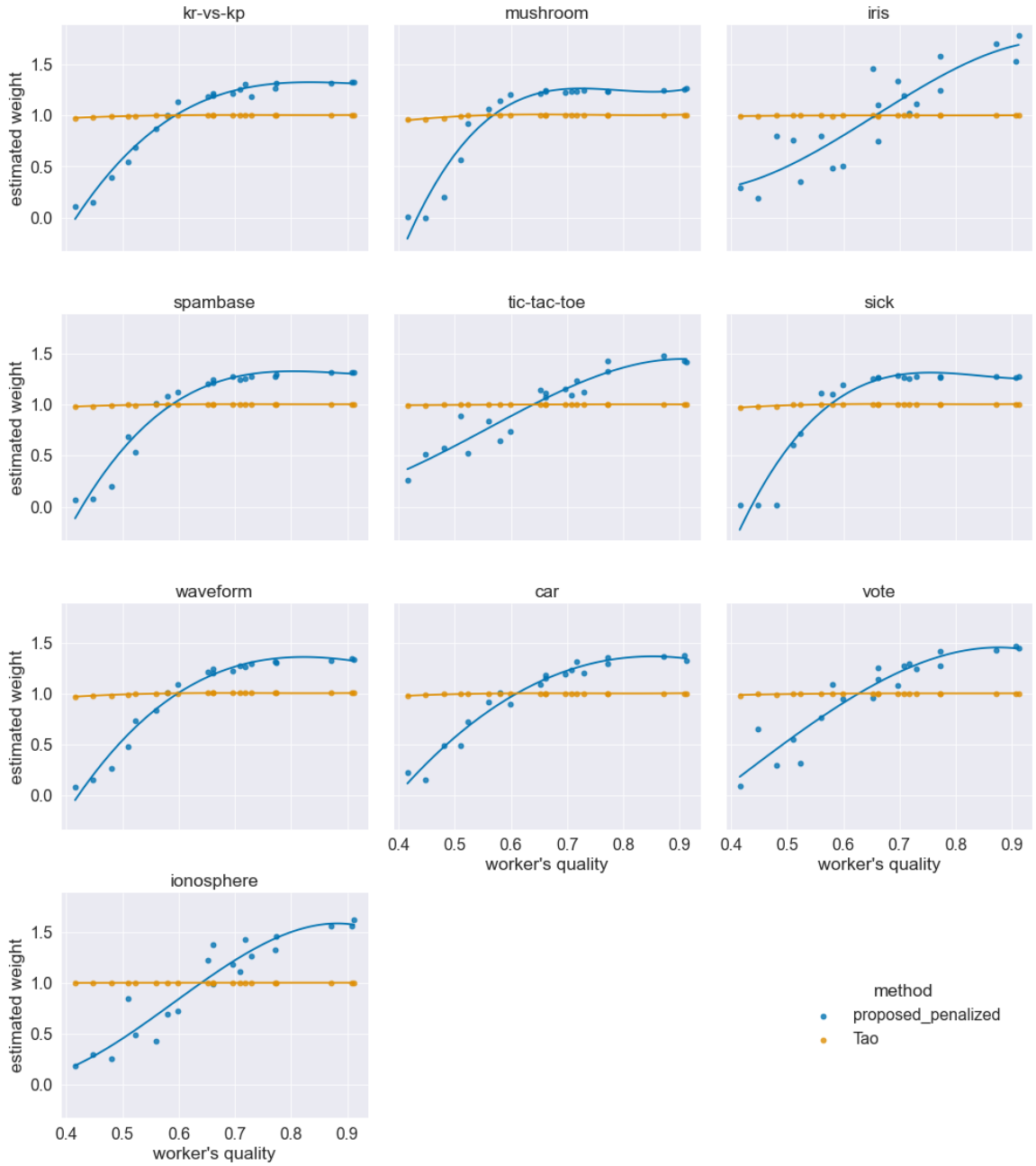

Figure 2.1. Comparison of estimated weight with respect to annotators' degree of reliability for the proposed aggregation technique "proposed-penalized" and Tao [**?**] for 10 different datasets.

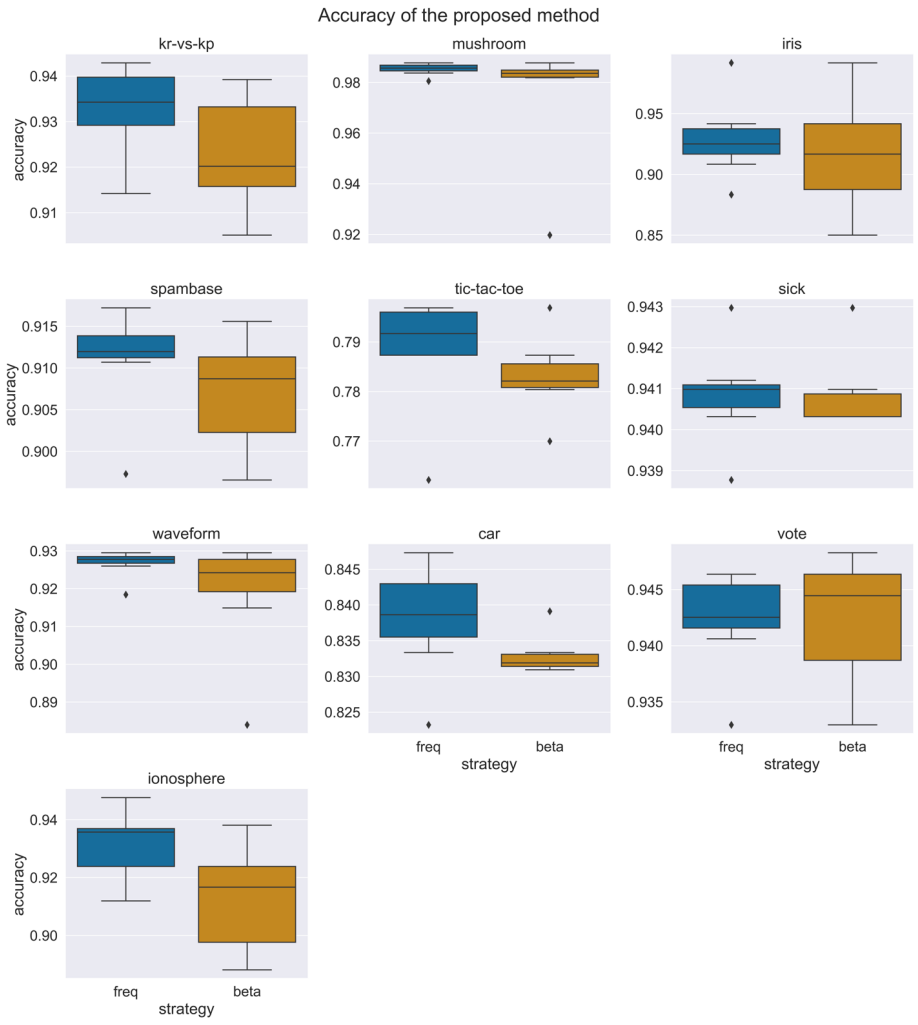

Figure 2.2. Comparison of the measured average accuracy for the two confidence-score measurement techniques in ten different datasets (using the proposed crowd-certain technique with penalization) in different numbers of annotators (from 3 up to 10).

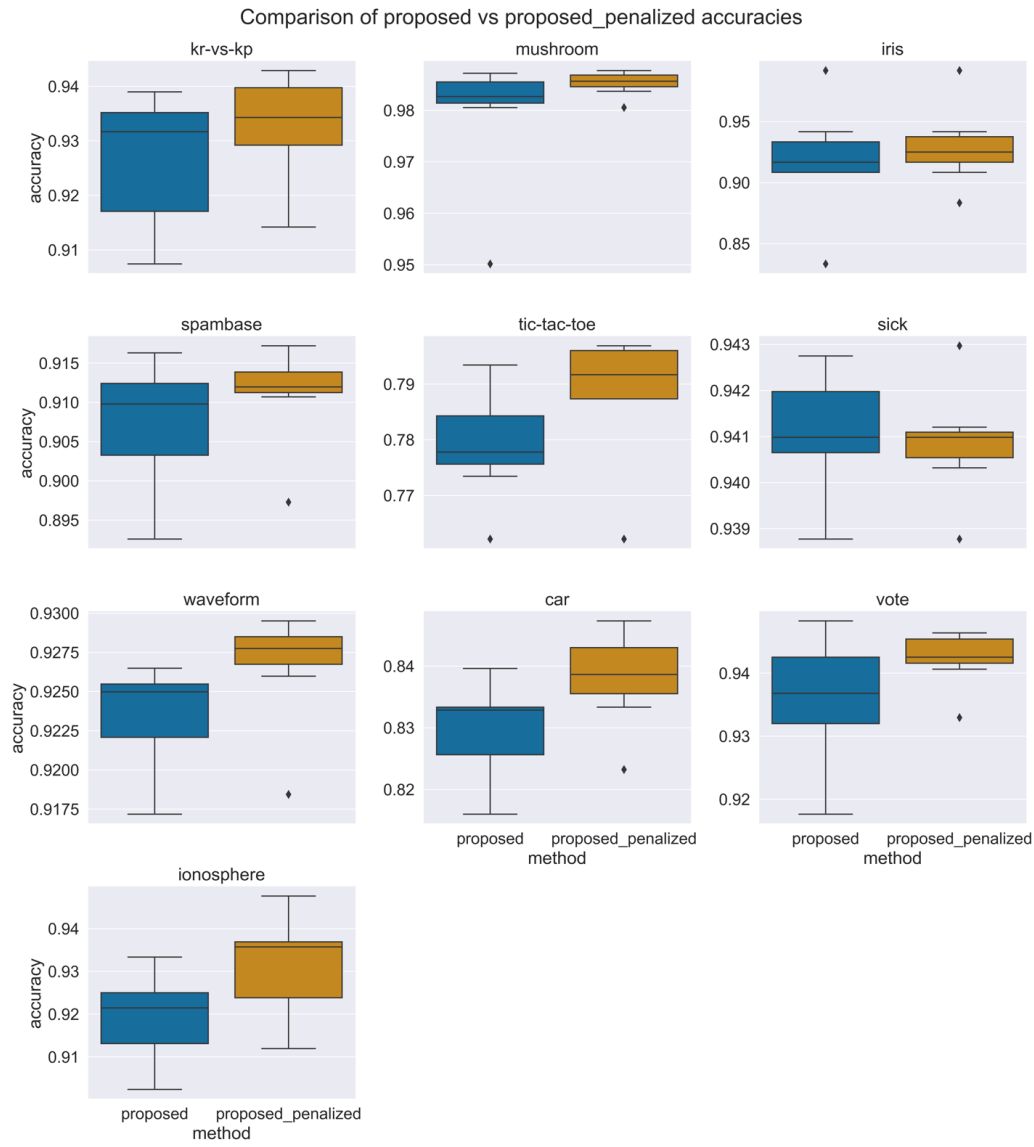

Figure 2.3. Comparison of the measured average accuracy for the proposed crowd-certain technique with and without penalization and using the 'freq' confidence-score strategy on ten different datasets with different numbers of annotators (starting from 3 to 10).

Table 2.2. Statistical tests between the proposed-penalized technique and Tao [**?**] for the 'freq' confidence measurement strategy.

Independent t-test	Diff	Degrees of freedom	t	2-sided p-value	Diff<0 p-value	Diff>0 p-value	Cohen d	Hedge's g	Glass's delta	Pearson r
Kr-vs-kp	-0.044	12	-6.171	0	0	1	-3.299	-3.088	-2.743	0.872
mushroom	-0.012	12	-2.781	0.017	0.008	0.992	-1.487	-1.392	-1.076	0.626
iris	-0.012	12	-0.506	0.622	0.311	0.689	-0.271	-0.253	-0.226	0.145
spambase	-0.031	12	-3.691	0.003	0.002	0.998	-1.973	-1.847	-1.458	0.729
tic-tac-toe	-0.036	12	-4.612	0.001	0	1	-2.466	-2.308	-2.156	0.8
sick	0.002	12	0.294	0.774	0.613	0.387	0.157	0.147	0.112	0.085
waveform	-0.025	12	-5.118	0	0	1	-2.736	-2.561	-2.022	0.828
car	-0.008	12	-0.779	0.451	0.226	0.774	-0.416	-0.39	-0.309	0.219
vote	-0.033	12	-5.352	0	0	1	-2.861	-2.678	-2.112	0.84
ionosphere	-0.061	12	-6.047	0	0	1	-3.232	-3.026	-2.586	0.868

density estimation technique. This demonstrates that the proposed technique outperforms both Tao and Sheng on nine of the ten datasets, with higher average accuracy and less fluctuation over different annotator counts. Table **??** shows the statistical data measured between the proposed penalized technique and Tao's method. As can be seen from these results, the proposed technique has a significant improvement over the seven datasets, while delivering similar results for the other three datasets (p-value < 0.05).

Figure **??** shows the average accuracy distribution ("Beta" strategy) of the proposed penalized versus Tao and Sheng strategies for different numbers of annotators using kernel density estimation. This demonstrates that the proposed technique outperforms both Tao and Sheng on seven out of ten datasets, with higher average accuracy and less fluctuation over different annotator counts. Furthermore, Table **??** shows the statistical data measured between the proposed penalized technique and Tao, showing a significant improvement in six datasets, while performing similarly in the remaining datasets (p-value < 0.05).

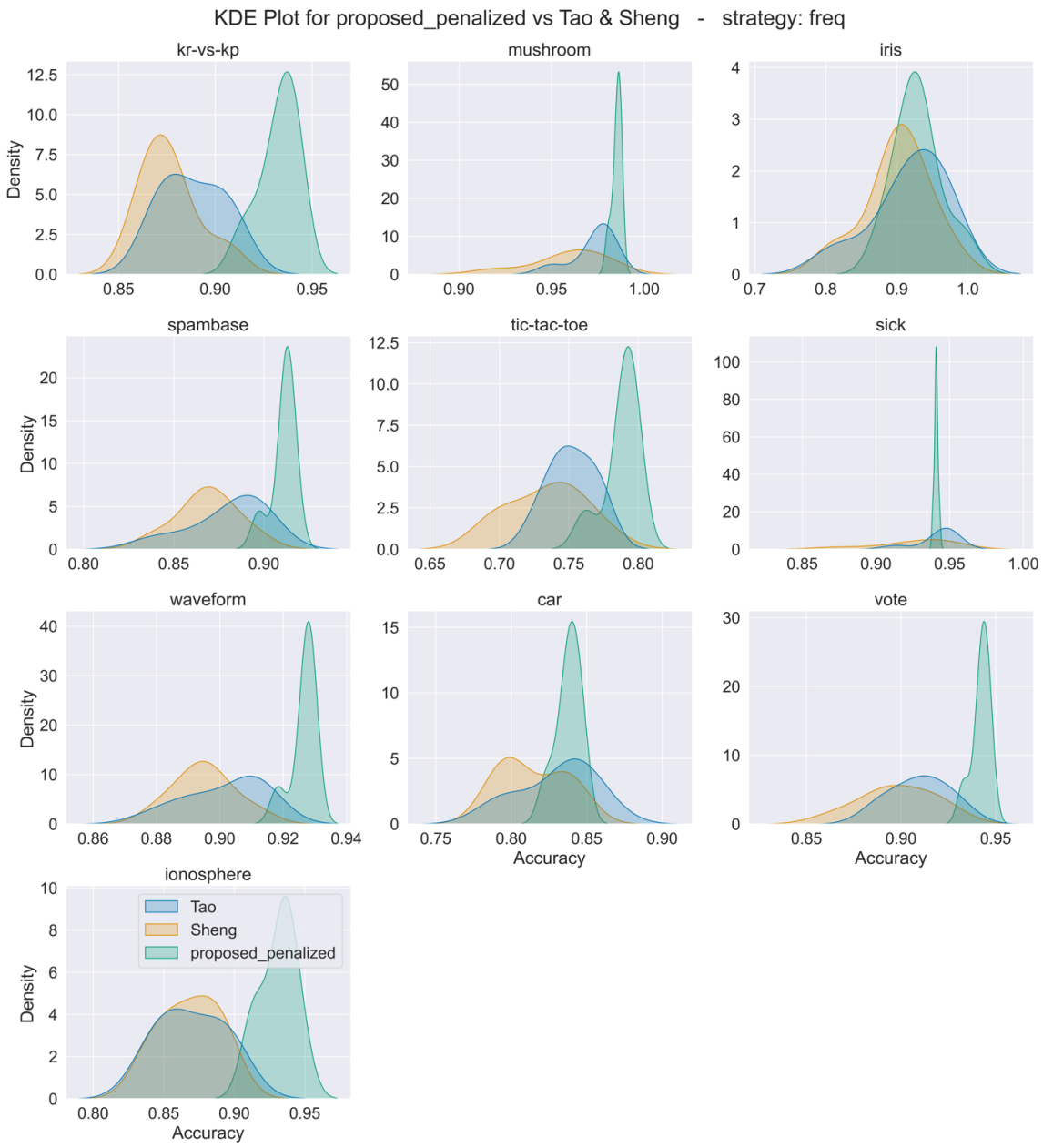

Figure 2.4. Measured accuracy distribution of the proposed-penalized aggregation technique uwMV-Freq, compared to wMV-Freq (Tao [**?**]), and MV-Freq (Sheng [**?**]) for different numbers of annotators, using the kernel density estimation technique

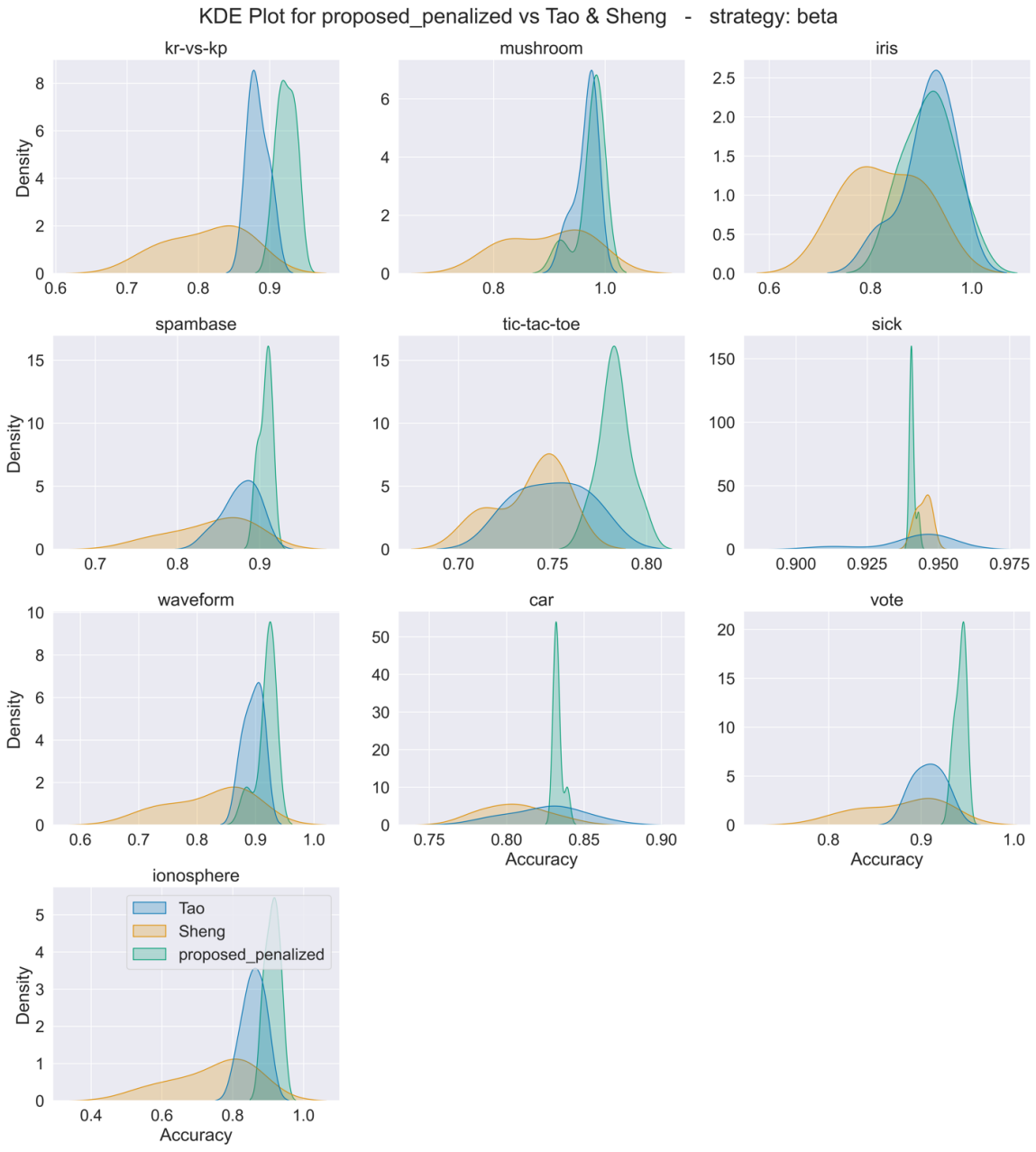

Figure 2.5. Measured accuracy distribution of the proposed-penalized aggregation technique uwMV-Beta, compared to wMV-Beta (Tao [**?**]), and MV- Beta (Sheng [**?**]) for different numbers of annotators, using the kernel density estimation technique.

Table 2.3. Statistical tests between the proposed-penalized and Tao [**?**] technique for the "Beta" confidence measurement strategy.

Independent t-test	Diff	Degrees of freedom	t	2-sided p-value	Diff < 0 p-value	Diff > 0 p-value	Cohen d	Hedge's g	Glass's delta	Pearson r
kr-vs-kp	-0.04	12	-5.702	0	0	1	-3.048	-2.854	-2.921	0.855
mushroom	-0.009	12	-0.793	0.443	0.222	0.778	-0.424	-0.397	-0.477	0.223
iris	-0.002	12	-0.091	0.929	0.465	0.535	-0.048	-0.045	-0.048	0.026
spambase	-0.03	12	-3.547	0.004	0.002	0.998	-1.896	-1.775	-1.421	0.715
tic-tac-toe	-0.033	12	-4.326	0.001	0	1	-2.312	-2.165	-1.781	0.781
sick	0.001	12	0.14	0.891	0.554	0.446	0.074	0.07	0.053	0.04
waveform	-0.023	12	-2.672	0.02	0.01	0.99	-1.428	-1.337	-1.442	0.611
car	-0.008	12	-0.871	0.401	0.2	0.8	-0.465	-0.436	-0.332	0.244
vote	-0.034	12	-5.198	0	0	1	-2.779	-2.601	-2.081	0.832
ionosphere	-0.052	12	-3.875	0.002	0.001	0.999	-2.071	-1.939	-1.742	0.746

Figure **??** shows the average accuracy for the "ionosphere" dataset for various annotator counts (horizontal axis). As can be seen, the proposed strategies considerably improve accuracy while utilizing a small number of annotators. Furthermore, Figure **??** shows the average accuracy of the three annotators (the smallest number of annotators that could perform consensus voting) on all ten datasets. Similarly, for the ionosphere dataset, we observed a similar trend in achieving higher accuracy on nine of the ten datasets compared to all benchmarks. It is important to note that the "freq" confidence measurement strategy is used to report the proposed techniques, as well as the Tao and Sheng results. Furthermore, the measured p-value calculated for the measured average accuracy (using three annotators) over different datasets showed a significant improvement for the proposed technique with and without penalization over all remaining benchmarks (Gold Majority Vote, MV, MMSR, Wawa, Zero-Based Skill, GLAD, Dawid Skene).

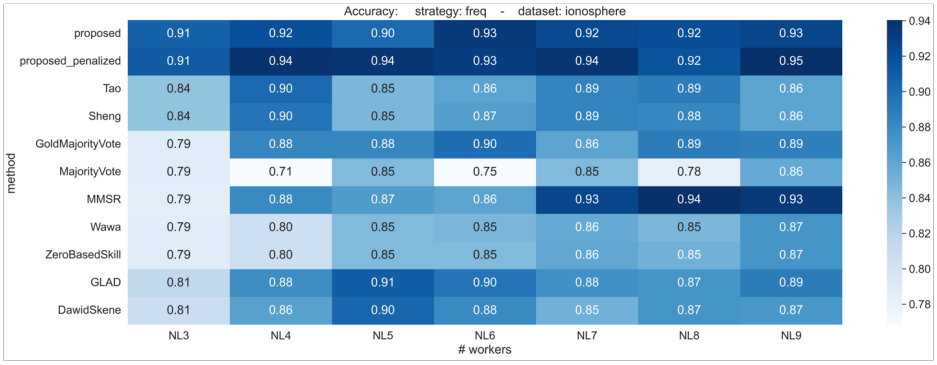

Figure 2.6. Average accuracy for the proposed aggregation techniques compared to the benchmarks for different numbers of annotators (horizontal axis) in the ionosphere dataset.

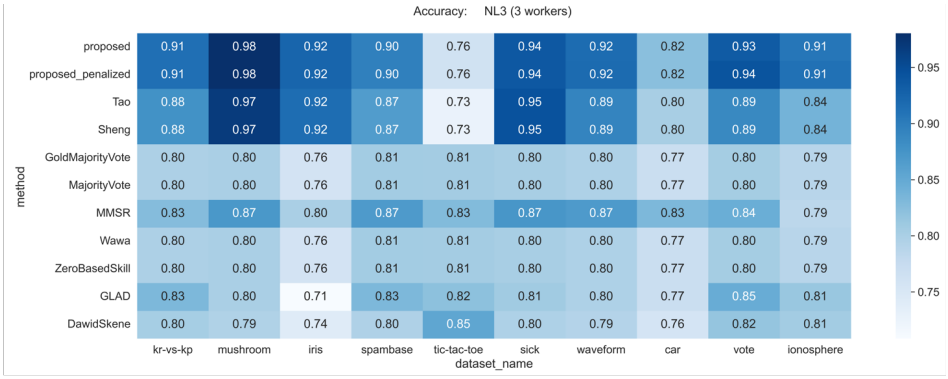

Figure 2.7. Average accuracy of the proposed aggregation techniques compared to the benchmarks for different datasets using three annotators.

2.5 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 measures annotator reliability based on their consistency and accuracy, in relation to other annotators. We utilized 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, we improved our initial strategy by penalizing annotator reliability estimates based on their inconsistencies in labeling. The first part of the proposed algorithm (calculating weights based on consistency) is essential because non-expert annotators often exhibit more irregular consistency during labeling than experts, as they are not trained to identify specific features. This measure helps to differentiate skilled and unskilled annotators. The goal of the second part of the algorithm (penalty for voting against the majority) 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 condition 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.

2.6 Availability of data and materials

The code can be found in crowd-certain

2.7 Appendices

List of abbreviations

Competing interests

Acknowledgements

References

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

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

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

[Bellaviti et al., 2016] Bellaviti, N., Bini, F., Pennacchi, L., Pepe, G., Bodini, B., Ceriani, R., D'Urbano, C., and Vaghi, A. (2016). Increased Incidence of Spontaneous Pneumothorax in Very Young People: Observations and Treatment. *CHEST*, 150(4):560A.

[Bi and Kwok, 2014] Bi, W. and Kwok, J. T. (2014). Mandatory Leaf Node Prediction in Hierarchical Multilabel Classification. *IEEE Trans. Neural Netw. Learning Syst.*, 25(12):2275–2287.

[Bi and Kwok, 2015] Bi, W. and Kwok, J. T. (2015). Bayes-Optimal Hierarchical Multilabel Classification. *IEEE Trans. Knowl. Data Eng.*, 27(11):2907–2918.

[Bustos et al., 2020] Bustos, A., Pertusa, A., Salinas, J.-M., and de la Iglesia-Vayá, M. (2020). Padchest: A Large Chest X-Ray Image Dataset With Multi-Label Annotated Reports. *Medical Image Analysis*, 66:101797.

[Cai et al., 2018] Cai, J., Lu, L., Harrison, A. P., Shi, X., Chen, P., and Yang, L. (2018). Iterative Attention Mining for Weakly Supervised Thoracic Disease Pattern Localization

in Chest X-Rays. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G., editors, *Med. Image Comput. Comput. Assist. Interv. – MICCAI 2018*, Lecture Notes in Computer Science, pages 589–598, Cham. Springer International Publishing.

[Chen et al., 2019] Chen, H., Miao, S., Xu, D., Hager, G. D., and Harrison, A. P. (2019). Deep Hierarchical Multi-Label Classification of Chest X-Ray Images. In *Proc. 2nd Int. Conf. Med. Imaging Deep Learn.*, pages 109–120. PMLR.

[Chen et al., 2020] Chen, H., Miao, S., Xu, D., Hager, G. D., and Harrison, A. P. (2020). Deep Hiearchical Multi-Label Classification Applied to Chest X-Ray Abnormality Taxonomies. *Medical Image Analysis*, 66:101811.

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

[Crisp and Chen, 2014] Crisp, N. and Chen, L. (2014). Global Supply of Health Professionals. *N Engl J Med*, 370(10):950–957.

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

[Dembczyński et al., 2012] Dembczyński, K., Waegeman, W., Cheng, W., and Hüllermeier, E. (2012). On Label Dependence and Loss Minimization in Multi-Label Classification. *Mach Learn*, 88(1-2):5–45.

[Dimitrovski et al., 2011] Dimitrovski, I., Kocev, D., Loskovska, S., and Džeroski, S. (2011). Hierarchical Annotation of Medical Images. *Pattern Recognition*, 44(10-11):2436–2449.

[Guan et al., 2018] Guan, Q., Huang, Y., Zhong, Z., Zheng, Z., Zheng, L., and Yang, Y. (2018). Diagnose Like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification.

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

[Guo et al., 2018] Guo, Y., Liu, Y., Bakker, E. M., Guo, Y., and Lew, M. S. (2018). CNN-RNN: A Large-Scale Hierarchical Image Classification Framework. *Multimed Tools Appl*, 77(8):10251–10271.

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

[Irvin et al., 2019] Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., Seekins, J., Mong, D. A., Halabi, S. S., Sandberg, J. K., Jones, R., Larson, D. B., Langlotz, C. P., Patel, B. N., Lungren, M. P., and Ng, A. Y. (2019). CheXpert: A Large Chest Radiograph Dataset With Uncertainty Labels and Expert Comparison. In *Proc. AAAI Conf. Artif. Intell.*, volume 33, pages 590–597.

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

[Jaderberg et al., 2015] Jaderberg, M., Simonyan, K., Zisserman, A., and kavukcuoglu, k. (2015). Spatial Transformer Networks. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Adv. Neural Inf. Process. Syst.*, volume 28. Curran Associates, Inc.

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

[Johnson et al., 2019] Johnson, A. E. W., Pollard, T. J., Berkowitz, S. J., Greenbaum, N. R., Lungren, M. P., Deng, C.-y., Mark, R. G., and Horng, S. (2019). MIMIC-CXR, a De-Identified Publicly Available Database of Chest Radiographs With Free-Text Reports. *Sci Data*, 6(1):317.

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

[Lakhani and Sundaram, 2017] Lakhani, P. and Sundaram, B. (2017). Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. *Radiology*, 284(2):574–582.

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

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

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

[Nguyen et al., 2022] Nguyen, H. Q., Lam, K., Le, L. T., Pham, H. H., Tran, D. Q., Nguyen, D. B., Le, D. D., Pham, C. M., Tong, H. T. T., Dinh, D. H., Do, C. D., Doan, L. T., Nguyen, C. N., Nguyen, B. T., Nguyen, Q. V., Hoang, A. D., Phan, H. N., Nguyen, A. T., Ho, P. H., Ngo, D. T., Nguyen, N. T., Nguyen, N. T., Dao, M., and Vu, V. (2022). VinDr-CXR: An Open Dataset of Chest X-Rays With Radiologist's Annotations. *Sci Data*, 9(1):429.

[Pasa et al., 2019] Pasa, F., Golkov, V., Pfeiffer, F., Cremers, D., and Pfeiffer, D. (2019). Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization. *Sci Rep*, 9(1):6268.

[Pourghassem and Ghassemian, 2008] Pourghassem, H. and Ghassemian, H. (2008). Content-Based Medical Image Classification Using a New Hierarchical Merging Scheme. *Computerized Medical Imaging and Graphics*, 32(8):651–661.

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

[Roy et al., 2020] Roy, D., Panda, P., and Roy, K. (2020). Tree-Cnn: A Hierarchical Deep Convolutional Neural Network for Incremental Learning. *Neural Networks*, 121:148–160.

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

[Tsoumakas and Katakis, 2007] Tsoumakas, G. and Katakis, I. (2007). Multi-Label Classification: An Overview. *Int. J. Data Warehous. Min.*, 3(3):1–13.

[Van Eeden et al., 2012] Van Eeden, S., Leipsic, J., Paul Man, S. F., and Sin, D. D. (2012). The Relationship Between Lung Inflammation and Cardiovascular Disease. *Am J Respir Crit Care Med*, 186(1):11–16.

[Wang et al., 2017] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017). 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*, pages 3462–3471, Honolulu, HI. IEEE.

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

[Zhang and Zhou, 2014] Zhang, M. L. and Zhou, Z. H. (2014). A Review on Multi-Label Learning Algorithms. *IEEE Trans. Knowl. Data Eng.*, 26(8):1819–1837.