Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Mani Sotoodeh Date

To my parents, Fereshteh & Safar, whose affection holds all love & caring can be.

Crowdsourcing and Semi Supervised Learning for Detection and Prediction of Hospital Acquired Pressure Ulcer Injury

By

Mani Sotoodeh
Doctor of Philosophy

Computer Science and Informatics

Joyce Ho, Ph.D.
Advisor

Li Xiong, Ph.D.
Co-advisor

Vicki Hertzberg, Ph.D.
Committee Member

Imon Banerjee, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Crowdsourcing and Semi Supervised Learning for Detection and Prediction of Hospital Acquired Pressure Ulcer Injury

By

Mani Sotoodeh
B.Sc., University of Tehran, Tehran, 2015
M.Sc., Emory University, GA, 2020

Advisor: Joyce Ho, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2021

Abstract

Crowdsourcing and Semi Supervised Learning for Detection and Prediction of Hospital Acquired Pressure Ulcer Injury
By Mani Sotoodeh

Pressure ulcer injury (PUI) or bedsore is "a localized injury to the skin and/or underlying tissue due to pressure." More than 2.5 million Americans develop PUI annually, and the incidence of hospital-acquired PUI (HAPUI) is around 5% to 6%. Bedsores are correlated with reduced quality of life, higher mortality and readmission rates, and longer hospital stays. The Center for Medicare and Medicaid considers PUI as the most frequent preventable event, and PUIs are the 2nd most common claim in lawsuits. The current practice of manual quarterly assessments for a day to estimate PUI rates has many disadvantages including high cost, subjectivity, and substantial disagreement among nurses, not to mention missed opportunities to alter practices to improve care instantly. The biggest challenge in HAPUI detection using EHRs is assigning ground truth for HAPUI classification, which requires consideration of multiple clinical criteria from nursing guidelines. However, these criteria do not explicitly map to EHRs data sources. Furthermore, there is no consistent cohort definition among research works tackling HAPUI detection. As labels significantly impact the model's performance, inconsistent labels complicate the comparison of research works. Multiple opinions for the same HAPUI classification task can remedy this uncertainty in labeling. Research works on learning with multiple uncertain labels are mainly developed for computer vision. Unfortunately, however, acquiring images from PUIs at hospitals is not standard practice, and we have to resort to tabular or time-series data. Finally, acquiring expert nursing annotations for establishing accurate labels is costly. If unlabelled samples can be utilized, a combination of annotated and unlabelled samples could yield a robust classifier. To overcome these challenges, we introduce the following: 1) Proposing a new standardized HAPUI cohort definition applicable to EHR data loyal to clinical guidelines; 2) A novel model for learning with unreliable crowdsourcing labels using sample-specific perturbations, suitable for sparse annotations of HAPUI detection (CrowdTeacher); 3) Exploration of unstructured notes for enhancement and gleaning better feature representations for HAPUI detection; 4) Incorporating unlabelled data into HAPUI detection via semi-supervised learning to reduce annotation costs.

Crowdsourcing and Semi Supervised Learning for Detection and Prediction of Hospital Acquired Pressure Ulcer Injury

By

Mani Sotoodeh
B.Sc., University of Tehran, Tehran, 2015
M.Sc., Emory University, GA, 2020

Advisor: Joyce Ho, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2021

Acknowledgments

I first want to thank my advisor, Dr. Joyce Ho, for allowing me find my way in the academic maze instead of bounding me to any particular research topic. I've learned so much from her, not only on how to creatively approach problems but also on how to present myself and my ideas better and prioritize my academic goals toward what I'd like to achieve the most while being realistic. Her tenacity in following up on ideas and not giving up on intermediate not-so-interesting results has taught and inspired me along the way. I'm also very grateful for her appreciation of my talents and skills and always encouraging me to promote them, be proud of my accomplishments, and never underestimate them. Having her as an academic advisor has made me feel safe and comfortable and supported all along my academic journey as she's been a truly caring academic mentor, and she has genuinely acted in my best interest. Lastly, I've also learned a lot from her about professional communication within academia and effective academic relationships, and connection-making. I find her a very smart, ambitious, punctual, respectful, and well-rounded researcher.

I also want to thank my Co-advisor, Dr. Li Xiong, for her endless curiosity, insistence, warmness, willingness to unravel all the whys, attention to details, and constant juxtaposition of how results measure up against one's intuitions. I've learned a great deal from her about effective academic writing, consistency in terminology and the power of the words, and proper academic style. She's also a great teacher, and her class was the most engaging and enjoyable class I took in my Ph.D. year. I appreciate all the good times I have spent with the AIMS group in lab outings which made me feel part of a community beyond academic collaborations. I feel lucky to have been part of such a supportive lab during my studies. I'm thankful to Dr. Vicki Hertzberg, Dr. Imon Banerjee, Dr. Wenhui Zhang and Dr. Roy Simpson for their insightful comments and discussions, making my research work stronger.

I also want to thank all the wonderful teachers I have had, from elementary school

till the end of college; Ms. Zahmatkesh, my 2nd grade teacher, for gifting me a book every week so that I read more, my first English teacher in 6th grade for encouraging me to improve my handwriting, Mr. Ashkboos my high school Arabic teacher for his engaging etymology stories, Prof Peyman Nasehpoor and Mahmood Shabankhah for being great instructors and their support when I applied to grad school. The faculty at the Computer science department and other graduate students, especially AIMS and PRADA lab members, LGS staff and CS department staff, have been essential to my PhD experience, and I appreciate their presence.

I'm deeply grateful for my parents' patience with me all these years away from them and home, their encouragement, and their belief in my abilities. Their love and affection have been a constant source of light and hope for me, even in the most challenging times. The riddles my mom asked me as a child and the variety of books she bought for me hooked me to science from an early age. My dad's presence in my school years for meetings and all the extracurricular classes they signed me up for and took me despite their busy lives has significantly contributed to my understanding of the world. Simply all my achievements would never exist without their support, care, time, and love for me.

I'm also forever in debt to my brother for always being there for me emotionally and letting me use his brainpower all these years. For always expressing I deserve more from what I already have, that my potential is much higher than I think and that everything will be alright. Thank you, Parsa, for listening to all my nonsense and complaints all these years and helping me make better decisions as I grew and found my way in the world.

I want to thank Farnaz for all the good times I had in her company and her advices these years. She's been a great friend ever since we became friends and was there for me most of my hard times. Farnaz and Azad were also great neighbors whose company and positive energy I enjoyed a lot. I also appreciate the presence of former

Iranian Atlanta friends who helped me better understand myself.

I'm grateful for all the Emory resources that kept me attached to this world and sane during my Ph.D. Lovely members of Wednesday Compassion Meditation Group, Emory Capoeira Club, staff and group members at CAPS interpersonal and men's group, staff at Emory office of LGBT life, especially people in the graduate queer discussion group and all the kind people at ISSS office and ESLP program. These all helped me see beyond myself and the limited knowledge I had of myself and enabled me to grow as a person.

I also want to thank Ayda, Ellie, and Devin, for existing and their sweetness, giving me a little burst of joy every time I see them, in real life or virtually.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Pressure ulcer injury (PUI) or bedsore is formally defined as "a localized injury to the skin and/or underlying tissue usually over a bony prominence or related to a medical or other devices, as a result of pressure, or pressure in combination with shear"[25, 49]. More than 2.5 million Americans develop PUI annually [8] and incidence of hospital-acquired PUI is estimated around 5% to 6% [8, 14, 67]. Bedsores are correlated with reduced quality of life, higher mortality, higher readmission rates (75% higher than other chronic conditions), longer hospital stays, more institutionalization after hospitalization, and economic burdens on patients and healthcare units ($500-7,000 for each PUI) [8, 15, 51, 58, 60, 73]. Center for Medicare and Medicaid (CMS) reports PUI as the most frequent preventable event, and PUI is also the 2nd most common claim in lawsuits after a wrongful death [5].

Hospital-acquired pressure ulcer injury (HAPUI), that is, PUI developing after initial admission evaluation, is a key metric for patient safety and a primary nursing quality indicator, illuminating the caliber of nursing expertise within a hospital [8, 51]. CMS and Agency for Healthcare Research and Quality (AHRQ) have both defined HAPUI quality indicators which should be periodically reported by care units [3, 7]. More specifically, from 2015, any HAPUI, is considered preventable, and Medicare pe-

nalizes the bottom 25% of the lowest-performing hospitals [17]. Accurate estimation of HAPUI incidence within any healthcare unit is vital for nursing quality assessment and proper planning by hospital administrative staff. Also, to avoid financial difficulties and increase their reputation, many healthcare institutions are trying to more accurately evaluate HAPUI risks to allocate resources such as specialized pressure mattresses for those cases and direct more nursing attention for monitoring skin status [44]. The current practice of manual quarterly assessments of one hospital in a single day by supervisor nurses for assuring/reporting quality metrics or informing nursing care through the estimated PUI rate has many disadvantages. High cost, subjectivity, and substantial disagreement among nurses [71], missed opportunities to instantly alter practices leading to inadequate care are all areas needing improvement in HAPUI detection. Thus, the need for an accurate HAPUI detection tool arises to inform nurses timely about appropriate interventions.

Electronic Health Records (EHRs) have recently been used to assist clinical decision-making, and patient care [11]. EHRs can be mined to detect HAPUI as they encompass multiple data sources indicating the development of a new or the presence of existing PUIs. More frequent PUI incidence estimation rates through EHRs has the potential to boost efficiency and outcomes for nurse-driven plans of care exponentially. For example, once HAPUI is detected through machine learning, the RN has the knowledge and power to direct caregivers in the tactical care of the patient for eliminating the adverse outcomes associated with HAPUI. From the perspective of efficient nursing quality assurance, the predicted incidence of HAPUI is a reliable proxy for its incidence in reality. In the presence of EHR-based HAPUI classifiers, to achieve more reliable HAPUI detection, we can present nurses with only the most uncertain cases for confirmation, rather than overwhelming them with all hospital stays.

Despite advances in clinical decision-making using EHRs, HAPUI detection has

not been sufficiently studied, and it still remains a challenging task. Risk assessment methods such as the Braden scale [12] has been used for its risk evaluation in the past. However, these classical methods have unstable sensitivity and specificity due to demographical drift [19], poor prognostic accuracy [21], and no apparent impact on decreasing HAPUI incidence [46]. More recently, proposed personalized risk algorithms have considered a broader range of patient-level factors than these earlier studies but are mainly using structured data from EHRs [20, 21]. Unstructured clinical notes contain often overlooked insights about HAPUI and should be one of the key data sources. Moreover, most of these works aim at exploring HAPUI patients to uncover risk factors retrospectively rather than detecting HAPUI development within a reasonable time interval, which is of higher clinical significance.

The most salient challenge in identifying HAPUI using EHRs is that assigning ground truth for HAPUI classification requires consideration of many clinical criteria dictated by respective guidelines [1, 2, 3]. However, these criteria often do not explicitly map to EHRs data sources, preventing accurate labeling and evaluation. There is no consistent cohort definition among research works that try to detect HAPUI [22, 28, 56], even though they are using the same dataset. HAPUI cohorts within these studies are divergent despite their similar claim of detecting HAPUI. Moreover, the labels significantly impact the model's performance to classify HAPUI, and inconsistent labels seriously complicate comparing these research works. Therefore, inferring HAPUI labels from EHR data and defining a clinically meaningful cohort based on regulatory guidelines is an essential task for any data-driven study for HAPUI detection.

However, even if we determine labels for HAPUI tasks as accurately as possible based on available EHR sources and nursing guidelines, there is still some inherent noise in these EHR sources. Additionally, clinical guidelines have some subjective components too. These two factors represent the second challenge in EHR-based

HAPUI detection, i.e., uncertain labels despite a standardized cohort. To tackle this challenge we resort to learning with uncertain crowdsourcing paradigms since this uncertainty in labeling naturally lends itself to a crowdsourcing problem: by obtaining multiple opinions for the same HAPUI classification task and leveraging them for learning, we can deal with the lack of reliable ground truth for the classifier. There are many research works focusing on learning with uncertain crowdsourcing labels, though mostly they are developed in conjunction with computer vision benchmark tasks such as MNIST and CiFAR100 [9, 59].

Although some researchers have tackled crowdsourcing for biomedical applications using typical hospital images [29, 66], acquiring images from PUIs at hospitals is not currently a standard practice and, therefore, we have to resort to tabular or time-series data in EHR for HAPUI detection, which renders these methods inapplicable. We can explore another tangentially related topic to crowdsourcing with unreliable labels, learning with noisy label paradigms. Data augmentation and selective gradient propagation are two examples of this approach that have achieved great success in robust learning despite noisy labels. Nevertheless, most of the proposed algorithms are benchmarked on Computer Vision or Natural Language Processing (NLP) tasks and cannot be directly used for tabular EHR data. Adopting these approaches to tabular data is nontrivial and challenging. For instance, data augmentation for images is routinely defined over standard transformations like rotations and resizing, but similar transformations for tabular data do not exist. Similarly, these approaches are highly dependent on training size, and generating healthcare data of the same scale requires stricter privacy guarantees and inter-organizational coordination to ensure integrated standardized EHR data across hospitals [31, 37, 79].

The third limitation in the current EHR-based HAPUI detection methods is the failure to utilize rich hospital notes. Existing models are developed on structured data [20, 21] and may not reflect all the patient information as hospital notes con-

tain critical information about the prognosis of HAPUI that might not be present in other structured EHR sources. Therefore exploration of notes in learning can significantly improve the performance and thus the adoption of an EHR-based method for detecting HAPUI.

The fourth challenge to developing an EHR-based HAPUI detection system is from a practical perspective; acquiring expert nursing annotations for establishing high-quality labels for HAPUI detection is costly. Therefore, it is unreasonable to require thousands of multi-annotated records on a real-world scale. However, if unlabelled samples can effectively contribute to model training, a combination of annotated samples in addition to unlabelled ones could yield a robust classifier.

Considering these four-fold challenges and limitations for HAPUI detection using EHRs, this dissertation presents the following four contributions:

- We propose a new standardized HAPUI cohort definition based on EHR data that examines all available data sources, including text, chart events and procedures, and diagnosis codes, unlike the current cohort definitions. Our cohort definition also incorporates clinical and regulatory guidelines, which have not been fully considered in the previous works.

- We introduce a novel model for learning with unreliable crowdsourcing labels using sample-specific perturbations, suitable for sparse annotations of HAPUI detection (CrowdTeacher). CrowdTeacher connects ideas from noisy labeling, synthetic sample generation, and crowdsourcing paradigms to fully harness the uncertainty throughout the training process and learn a genuinely robust classifier with sparse uncertain annotations, which has not been proposed before. CrowdTeacher outperforms the methods designed separately for each paradigm on synthetic, HAPUI, and length of stay classification datasets in most sparsity settings, demonstrating its generalizability.

- We explore unstructured notes for CrowdTeacher enhancement and gleaning better feature representations for HAPUI detection. We are the first to consider unstructured notes for HAPUI detection and further combine it with negation detection and confirm its utility both computationally and in terms of interpretability and compatibility with nursing knowledge.

- We leverage unlabeled data to reduce annotation costs for HAPUI detection via a modified self-training algorithm for the first time. We propose a modified self-training model combined with Co-teaching that is uniquely designed for imbalanced classes and classifiers trained on noisy labels. We showcase its success over vanilla self-training or Co-teaching alone on both HAPUI and length of stay classification tasks.

We dedicate the first section to an overview of the related work and necessary background. Afterward, in each chapter corresponding to the four projects, we provide details of motivations, challenges, and governing experimental settings, accompanied by extensive experiments and analysis illustrating their efficacy compared to existing methods. Finally, we summarize how each project fits with the others toward the holistic goal of an efficient EHR-based HAPUI detection paradigm.

Chapter 2

Background and Related Work

We provide a summary of previous literature relevant to our proposed work. We highlight the most relevant research works on truth inference in crowdsourcing, learning in the context of crowdsourcing, synthetic data generation, selective gradient propagation and semi-supervised learning. We also formally define multiple subproblems relevant to HPUI detection and detail their methodology.

2.1 Truth Inference in Crowdsourcing

Truth inference aims at inferring the most likely labels for crowdsourcing tasks, given the conflicting multiple annotations for each task.

2.1.1 Truth Inference Definition

Consider a crowdsourcing system consisting of some samples and a pool of annotators. Samples might be assigned to a subset of annotators. Truth inference aims to determine the true label of samples based on available annotation for each sample.

Given that HAPUI detection is a binary classification, only binary truth inference methods are discussed here. Examples of other binary crowdsourcing problems are

Annotator\ Sample	Annotator 1	Annotator 2	Annotator 3
Sample 1	-1	1	1
Sample 2	0	0	-1
Sample 3	1	-1	0
Sample 4	1	1	-1

Figure 2.1: Answer Matrix example.

determining the sentiment of sentences as positive or negative or reporting traffic conditions on the road using smartphones.

Definition 2.1.1. Truth Inference [57]: Suppose there is a set of R annotators $\mathbf{L} = \{l_1, .., l_R\}$ and N samples $\mathbf{S} = \{s_1, s_2, ..., s_N\}$ where each sample s_j has the truth label $z^*_{s_j} \in \mathbf{K}$. $\mathbf{K}$ is the set of all possible labels. Provided an answer matrix $\mathbf{A}$ where each element $a^{l_i}_{s_j}$ shows the answer from annotator l_i for sample s_j, the truth inference goal is to infer the true label $\hat{z}_{s_j} \in \mathbf{K}$ for each sample $s_j \in \mathbf{S}$. If sample s_j has not been assigned to annotator l_i , $a^{l_i}_{s_j} = -1$

Figure 2.1 depicts a sample crowdsourcing system comprised of 3 annotators and 4 samples for binary classification. The truth inference method's input is an answer matrix $\mathbf{A}$ provided by the annotators. Annotators label samples with 0 or 1, while -1 shows that the task was not assigned to that particular annotator. The output is $\hat{\mathbf{Z}}$, the inferred labels for the samples.

2.1.2 Truth Inference Methods

Four main classes of truth inference methods are 1) direct computation, 2) optimization, 3) probabilistic graphical model (PGM), and 4) neural networks. Direct computation learns from annotators' answers by majority voting and treats annotators equally or heuristically computes trust weights for them [33]. Optimization-based methods [33, 38, 39, 80, 82] consider the estimated labels and reliability of annotators as unknown variables and employ an optimization approach to estimate them. Probabilistic graphical models (PGM) explicitly model the reliability of annotators to

approximate the truth labels [23, 24, 35, 36, 68, 72]. Optimization and PGM-based methods incorporate an iterative Expectation-Maximization (EM)-based algorithm consisting of 1) inferring the samples label given the currently estimated reliabilities, and 2) recomputing reliability of the annotators given the current samples' inferred labels. More recently, unsupervised neural network-based approaches [27, 76] have been suggested that use as input each task's answers in a neural network and generate the samples' inferred label. Other methods rooted in tensor augmentation and completion with limited performance have been proposed too [81].

Confusion Matrix: The confusion matrix, π^{l_i}, captures reliabilites of annotators across all classes. π^{l_i} is a $|K| * |K|$ matrix in which element $\pi^{l_i}_{p,q}$ shows the probability of annotator l_i giving label q provided that the true label is p. Supposing a binary classification problem with label set $\mathbf{K} = \{0, 1\}$, then the number of matrix variables is reduced to only two, α_i and β_i, with $\alpha_i = \text{pr}(a^{l_i}_{s_j} = 1 \mid z^* = 1)$ and $\beta_i = \text{pr}(a^{l_i}_{s_j} = 0 \mid z^* = 0)$, showing the probability of annotator l_i truthfully reporting a sample given its true label being 1 or 0 respectively.

Two recent surveys have concluded that the D&S truth inference method [23] provides the best trade-off between computational efficiency and robustness and accuracy [40, 65]. Therefore we employ this method for the rest of the experiments in this thesis.

D&S truth inference method: The D&S [23] algorithm leverages EM to solve maximum likelihood estimation (MLE) for the inferred labels $\hat{\mathbf{Z}}$ and the confusion matrices $\pi^{\mathbf{L}}$ in an iterative manner. If $\mathbf{L}^{s_j}$ is the set of annotators who have been assigned to sample s_j, then the objective function of this method is: $\max_{\hat{\mathbf{Z}}, \pi^{\mathbf{R}}} \prod_{j=1}^{N} \sum_{k \in \mathbf{K}} pr(\hat{z}_{s_j} = k) \prod_{l_i \in \mathbf{L}:^{s_j}} \pi^{l_i}_{k, A^{l_i}_{s_j}}$.

2.2 Learning in the Context of Crowdsourcing

There are many applications in the real world in which the ground truth of a classification task is not available or conflicted. For example, in medicine, multiple pathologists do not always necessarily agree on the malignancy status of a tumor in an image [45], or multiple nurses with different backgrounds and experiences might not all agree on the presence of hospital-acquired bedsores for a patient given their charts [71]. Similarly, acquiring ground truth from experts to train reliable classifiers can be expensive, as in the case of content filtering and regulation of posts on social media, which are assigned to multiple non-expert annotators in order to obtain high-quality labels [53]. Formally, we define learning with crowdsourcing labels as follows:

Definition 2.2.1. (Classification with crowdsourcing Annotations) Consider a set of R annotators labeling N samples with K possible classes. Given an answer matrix $\mathbf{A} \in \mathbb{R}^{N \times R}$ where each element a_{nr} indicates the label for sample n provided by annotator r, and the training feature matrix $\mathbf{X}_{tr} \in \mathbb{R}^{N \times M}$, the goal is to train a classifier that accurately predicts the true labels for the test data using only its feature matrix $\mathbf{X}_{ts}$.

Classification with noisy answers or multiple crowdsourced labels overlaps with three other areas: learning with crowdsourcing labels, data augmentation and synthetic data generation for robust learning, and selective gradient propagation.

2.2.1 Paradigms in Learning with Crowdsourcing Labels.

Learning a classifier from crowdsourced labels has been studied from 3 perspectives. Here we summarize these high-level approaches for learning with multiple annotations.

Sequential. This approach first uses a truth inference method to approximate the ground truth for training samples. The estimated label is then used to train a classifier.

Simultaneous. The second perspective jointly deals with the problem of learning classifier parameters and the estimated ground truth of the samples. Albarqouni et al. employ the Expectation-Maximization (EM) algorithm and Maximum a posteriori estimation to iteratively compute these two collections of parameters until convergence [9]. Yet, this method is computationally restraining, especially for more complex classifiers and a large number of samples.

Individual annotator's label modeling. The last approach entail learning a model for each individual annotator. Dr. Net was proposed to learn a classifier to generate the labels of each annotator and is composed of two phases, individual annotator modeling and learning labelers' averaging weights for the final prediction [29]. To better handle the computational challenge of simultaneous learning and Dr. Net, multiple crowd-layer variants were introduced to remove the long runtime burden due to the EM loop [59], by first approximating the ground truth of samples and then attempting to replicate the individual annotator's labels using a very simple neural network. Unfortunately, such models require significant samples to properly learn a robust classifier.

2.3 Synthetic Data Generation

To combat the obstacle of noisy labels or features, perturbation schemes and data augmentations methods have been proposed. Perturbation of model parameters and architecture has proven to provide resiliency against noisy inputs in deep neural networks [10]. Perturbing input space through estimation of the distribution of features and augmenting the original sample points with multiple versions of such perturbations has also shown great success in achieving robustness in the face of noisy data [43, 78]. In computer vision, data augmentation is achieved through applying transformations like cropping and rotation to combat potential mislabelled training data

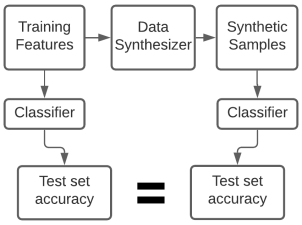


Figure 2.2: General framework of synthetic data generation.

[13, 61, 79]. Another line of work approaches robustness against noisy data by generating data synthesizers that achieve the same predictive performance as using real data. Xu et al. have extended data augmentations to tabular data with heterogeneous feature types using Generative Adversarial Networks and Variational Autoencoders [74]. Despite their success, unfortunately, such synthesizers are modeled independently of the labels or the conflicting annotations. Figure 2.2 shows the goal and the flow of this approach.

2.4 Selective Gradient Propagation and Co-teaching

The Co-teaching algorithm adaptively changes both the number of and the set of participating samples used in stochastic gradient descent epochs for two differently initialized classifiers to counter noisy labels and memorization effects in neural networks [31]. This algorithm modifies the samples fed to the network, with the logic of first feeding clean certain samples and gradually adding the more noisy ones as the classifier gets better at prediction. To distinguish cleans samples, these methods use the sample's associated loss as a proxy for its noisiness. Initialization of neural networks can generate different classifiers; therefore, to improve the overall generalizability, these methods use two classifiers with the same architecture but different initializations. For each epoch, Co-teaching chooses a different number of samples with the lowest loss (as a proxy for clean data) and updates each classifier using

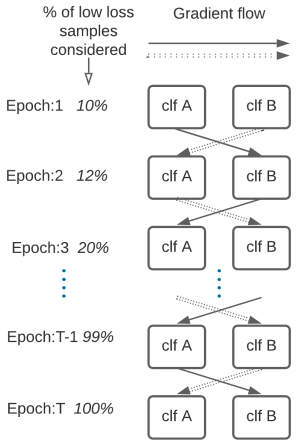


Figure 2.3: General framework of Co-teaching.

the clean samples of the other network. This is in contrast to the usual practice of using all the samples or the clean samples of the classifier itself, which may result in memorization and early overfitting that inhibits learning a generalizable and robust classifier. Either one of the classifiers can be used at test time due to the convergence. A parallel can be drawn to similarly deal with the inherent noisiness of aggregated crowdsourcing labels. The Co-teaching mechanism of prioritizing a smaller set of confident samples in the initial stages of learning and gradually incorporating more of the uncertain samples in later epochs can be employed for the problem of classification with crowdsourcing labels. Figure 2.3 depicts the flow of gradient of a Co-teaching framework with sample parameters.

2.5 Self-training, a Semi-Supervised Learning Paradigm

Self-training is one of the most intuitive and earliest semi-supervised methods [55]. In self-training, the classifier is iteratively trained on an increasingly larger set of labeled data. Given a labeled set, (X_L, y_L), the classifier is trained on X_L, for each iteration. The classifier prediction on an unlabelled set, X_U is then utilized to extend the labeled set by choosing a subset of the most certain samples from the unlabelled

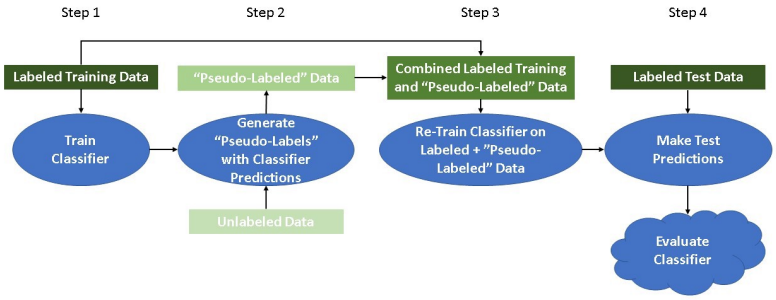


Figure 2.4: Basic schema of Self-training.

set and adding it to the already labeled set, (X_L, y_L). The criteria for the certainty of predictions for unlabeled samples vary, and the most common ones are choosing the top k most certain predictions from all classes and selecting samples whose certainty is more than a specified threshold. A simple schema of the self training algorithm is shown in Figure 2.4.

Some of the known disadvantages of self-training are its sensitivity to the quality of the base classifier and its inability to correct already misclassified samples in the labeled set added in the previous iterations [75].

Chapter 3

A Standardized HAPUI Cohort Compatible with Clinical Guidelines

EHRs' data sources, both structured and unstructured, can indicate HAPUI during a hospital stay. These sources often do not agree on the existence or the stage of HAPUI. However, making administrative decisions based on the HAPUI incidence, either standalone or in conjunction with output from a computational model, requires the cohorts and their labels to be well-defined and justified clinically. Moreover, reasonable class assignment criteria permit meaningful comparison of the computational models. In this chapter, we highlight the diverging nature of HAPUI sources within electronic records, provide reasons for part of their conflicts or overlappings, and suggest a new cohort for establishing HAPUI classification labels that more accurately follows the clinical guidelines compared to the current cohort definition. Finally, we analyze the performance of all cohorts on two types of classifiers.

EHRs can identify HAPUI as they include multiple data sources indicating the development of a new or existing PUI. These sources are codes; itemized PUI-related

events such as staging, depth, or location of PUI; and keywords extracted from semi-structured or unstructured clinical notes documenting the existence of a PUI and/or describing its features. Furthermore, multiple factors can impact the reliability of PUI's documentation: inherent complexity and subjectivity of its detection, screening, and staging; change in the nursing team composition within a hospital stay (i.e., continuity of care, nurse roles, and competencies); protocol changes for data entry, and changes to the EHR system. These might result in contradictory information across the data sources (e.g., regarding the presence of PUI). As a result, for the same dataset and task, there have been different cohort proposals. Therefore, defining the HAPUI cohort is a function of one's trust in each source's data quality and measurement reliability. Moreover, to genuinely assess the different machine learning models for HAPUI detection on a fixed benchmark data (such as the publicly available MIMIC-III dataset [34]) requires consideration of complex clinically justified criteria that combines multiple data sources consistent with the regulatory guidelines [3, 7].

We illustrate the challenges in detecting HAPUI in EHRs using MIMIC-III [34] as a case study. MIMIC-III is one of the most widely used open benchmark datasets, built over CareVue and Metavision EHR systems that encompass about 59k stays. It contains multiple tables for hospital stays, linking their vital signs, lab results, physiological measurements, demographics, notes, and diagnosis. Given the clinical context governing hospital stays, we employ nursing expertise to resolve some conflicts. We also categorize the required decisions to properly define a cohort for HAPUI detection in EHRs. Additionally, we evaluate the impact of existing cohort definitions on practical considerations such as generalizability and confidence in labels; and the effects of the definition on various computational models' performance. Finally, the strengths of our proposed cohort definitions for HAPUI classification are discussed. In summary, we aim to:

- Depict the challenges of finding evidence for HAPUI within different sources in

the MIMIC-III dataset.

- Provide clinical reasoning to reduce some of the conflicts in data sources.
- Define core parameters for a meaningful HAPUI cohort construction.
- Showcase the cohort definition effects on the performance of tree-based and neural network-based classifiers as two representative classifiers.

3.1 PUI Terminology in EHR

Here we define some terms used throughout the paper and their usage in EHRs.

HAPUI criteria. HAPUI is determined by regulatory authorities using many factors [3, 7], such as PUI admission and discharge stage, whether the patient was recorded deceased, changes in staging, and unit transfers during the stay. The CMS guideline provides the following six criteria:

1. Patient stays for which the discharge assessment indicates the presence of one or more new or worsened PUI (Stage 2–4, or unstageable pressure ulcers due to slough/eschar, non-removable dressing/device, or deep tissue injury) compared to admission, should be considered.
2. Patient stay is excluded if the patient died during the stay.
3. Patient stay is excluded if data on new or worsened Stage 2, 3, 4, and unstageable pressure ulcers, including deep tissue injuries, are missing on the planned or unplanned discharge assessment
4. If on admission a PUI was unstageable but becomes and remains numerically stageable later in the patient's stay, it should be coded as present on admission on the Discharge assessment at the stage at which it first becomes numerically stageable.

5. If a patient is transferred from the post-acute care (PAC) setting to an acute care hospital and returns within 3 days (including the day of transfer), the transfer is considered a program interruption and is not considered a new admission. Therefore, any new PUI formation or increase in numerical staging that occurs during the program interruption should not be coded as "present on admission."

6. The general standard of practice for newly admitted patients is that patient clinical admission assessments are completed as close to the actual time of admission as possible and usually within 24 hours.

AHRQ also recommends "ensure performance of comprehensive skin assessment within 24 hours of admission" to accurately measure PUI rates [1]. NPIAP reference guide [7] further defines facility acquired rate, Quality indicator 20, as "percentage of individuals who did not have a pressure injury on admission who acquire a pressure injury during their stay in the facility. "

Discharge ICD-9 codes. For billing purposes, a limited set of ICD-9 codes is chosen for each hospital stay at discharge. These codes' composition usually shows the most salient diagnoses made throughout the hospital stay. However, financial concerns and the imperfect mapping of clinical findings to these codes may violate the importance rule.

Clinical notes. For each hospital stay, numerous unstructured text might be accessible. These contain but are not limited to radiography reports, ECG and EKG reports, discharge summaries, admission notes, and daily notes made by the care team. We refer to the collective set of these textual data as clinical notes or notes for short.

Chart events. Chart events constitute the main portion of structured clinical data and include various medical services, such as lab tests, vital signs, nursing assessments, and general markers like the mental status of a patient. Chart events are almost

always time-stamped and give helpful information on a hospital stay's clinical events' time and order.

3.2 Challenges in PUI Cohort Definition

To highlight the existing limitations and justify the need for improvement for achieving our goal of developing an EHR-based classifier to identify HAPUI, we introduce the current cohort definitions in MIMIC-III. We then explore PUI data sources in MIMIC-III, showcase their conflicts, and provide some resolutions. We then analyze the impact of cohort definition for HAPUI classification using two types of classifiers on MIMIC-III.

3.2.1 HAPUI Detection from Multiple EHR Sources

Existing studies that leverage EHR data attempt to explore three perspectives:

- Explaining the most important indicators of HAPUI, such as particular comorbidities [20, 22, 50].
- Detecting the presence of HAPUI given all the records of a hospital stay [20, 21, 56]
- Leveraging earlier temporal lab data to predict HAPUI [28].

We compare our proposed cohort to existing cohorts for HAPUI classification/detection. The constructed cohort by [20] was not replicated as their cohort inclusion criteria were not clear, and [56] uses private local data.

CANTRIP[28]. This work attempts to predict HAPUI 48-96 hours before its first appearance, defined as DOE (Date of Event).

Cramer[22]. This study aims at giving an estimate of HAPUI incidence and identifying HAPUI cases. The cohort is labeled using only charts of stays.

3.2.2 Sources for PUI Hospital Stays in the MIMIC-III and Their Reliability

PUI is documented in MIMIC-III across four different tables: chartevents, noteevents, diagnoses-ICD, and CPT tables, and these tables are used to establish the labels for cohort hospital stays. We further pulled data from the Admissions, Patients, and ICU stays tables for our features.

Chartevents data. Each timed event in this table pairs an itemid with a medical concept. We first selected all PUI-stage-related events with itemids in Definition 3.2.1 as they are related to PUI staging concepts. For example, itemid '224970' is paired with 'Pressure Ulcer Stage #7'. This list is generated by taking the union of the CANTRIP and Cramer cohorts (with only an overlap of around 30% for HAPUI cases), since they both utilize chartevents to determine the label. For each specific event (hospital stay and time), the value attribute may be a number between 1-4 or other non-numerical values such as 'Unable to assess.' In the next step, we assign a stage numeral to events based on their 'value' attribute and the mapping based on [6]. There is also a 'valuenum' integer attribute for stage, but since it has many missing values, it isn't used. Note that hospital stays may have multiple PUI staging events.

Definition 3.2.1 (PUI-staging itemids in chartevents)**.** {**551**, **552**, **553**, **224631**, **224965**, **224966**, 224967, 224968, 224969, 224970, 224971, 227618 227619}

**only bold itemids are used by Cramer cohort*

ICD-9 Diagnosis data. We marked each hospital stay for the presence of PUI using the ICD-9 codes in Definition 3.2.2 based on [4], guaranteeing a PUI case. Unlike a previous study [28], we excluded ICD-9 codes that are regarded as non-pressure ulcers in the ICD-10 system.

Definition 3.2.2 (PUI-indicative ICD-9 codes)**.** {707, 707.1, 707.2, 707.3, 707.4, 707.5, 707.6, 707.7, 707.9, 707.11, 707.21, 707.22, 707.23, 707.24, 707.25}

Notes. Notes of each stay are checked for the existence of any of the keywords or regex patterns in Definition 3.2.3. This list covers most terms used to refer to PUI, including common misspellings, and disregards structural matches, i.e., "bed sore: none." as suggested in [28].

Definition 3.2.3 (Keywords and Regex patterns indicating PUI)**.** {bed sore($|[^ :]), bed ulcer($|[^ :]), pressure sore($|[^ :]), pressure ulcer($|[^ :]), decub($\backslash w * \backslash s*$) sore($|[^ :]), decub($\backslash w * \backslash s*$) ulcer($|[^ :])}

PUI-related CPT codes. CPT codes can indicate procedures to treat PUI, such as skin surgeries and debridements.

Definition 3.2.4 (PUI ICD-9 Codes)**.** {11042, 11043, 11044, 15999, 97598, 97597, 16020, 16030, 15835, 15878, 15879, 27027, 27057}

The conflict of these data sources for identifying MIMIC-III PUI case hospital stays without considering the patient attributes or time limitations is shown in Figure 3.1. This plot shows that close to 50% of the patients with any sign of PUI in their records only have them in their chartevent, while 10% only have a mention of PUI in the notes. The data sources' mismatch illustrates the need for reconciliation of these data sources and careful cohort definitions for any HAPUI classification task. Notes cannot be the only PUI case source since it results in false positives as the keywords can be preceded by negation (i.e., no PUI) and/or serve as suggestions to the patient to prevent PUI developments. Since ICD-9 codes focus on the most prominent ones due to the length limitation and might include previous conditions diagnosed before the hospital stay, they yield poor predictive performance on PUI tasks. Furthermore, the lack of a time-stamp makes them incompatible with HAPUI criteria.

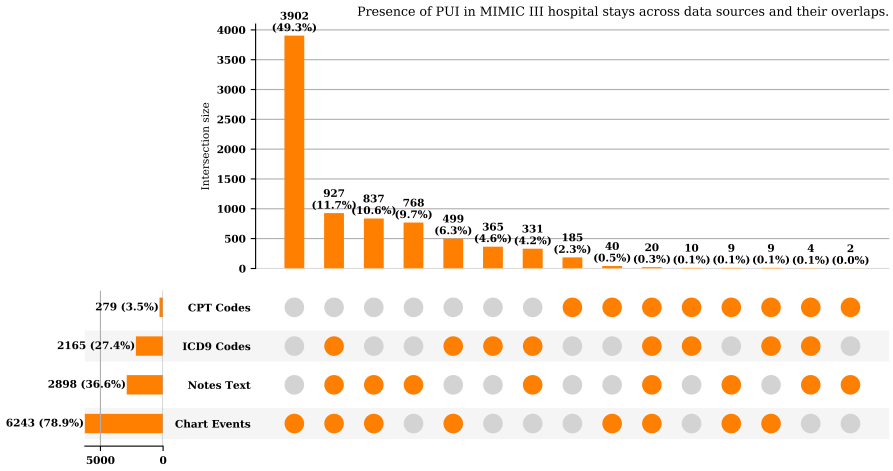


Figure 3.1: Conflict of PUI sources in MIMIC III Dataset.

3.2.3 The Existing Cohort Definitions v.s the New Cohort Definition

Based on the existing cohorts and the CMS HAPUI criteria, there are nine questions relating to demographics and detection of PUI cases that need to be answered to construct a cohort generally and for MIMIC-III specifically given its limitations.

Demographical Parameters.

- Q1: Do you exclude deceased patients?
- Q2: What is the minimum age to be included in a cohort?

HAPUI Case Criteria.

- Q3: What is the cutoff period (i.e., 24 or 48 hours after admission) beyond which PUI is associated with the current hospitalization?
- Q4: What are the set of chartevents itemids used to determine PUI staging data?

- Q5: What numerical stages should be used for deep tissue injury and unable to stage values?
- Q6: Should notes be used in addition to chart events staging data to establish labels?
- Q7: Do you include healed PUIs or PUIs which have gotten better by discharge?
- Q8: Should ICD-9 codes be used to select HAPUI cases?
- Q9: What is the minimum numerical stage for HAPUI?

CANTRIP cohort [28]. In contrast to HAPUI criteria, this cohort includes deceased patients, considers healed PUIs or PUIs that have gotten better during the stay, and only uses staging at a single time point to determine HAPUI cases. Also, the requirement for HAPUI cases is less strict as it allows stage 1, unconditioned deep tissue injury, and unable to stage values. Furthermore, it considers 48 hours instead of 24. If a stay has a staging of 1 or above, including deep tissue injury or unable to stage that happens later than 48 hours of admission, it is detected as a PUI case. Otherwise, if a note with a time later than 48 hours from admission has List 3 keywords, a PUI case is determined. All other stays fall into the control samples category.

Cramer cohort [21]. This cohort only uses the staging chartevents data with a more limited list of itemids to determine PUI cases. It also does not conform to the HAPUI criteria as it considers dead patients and uses staging information from a single time point. For each stay, if there is a staging of 2 or above, excluding unable to stage and deep tissue injury that is timed later than 24 hours of admission, it is marked as a HAPUI case. All the other stays make up the control population.

Golden cohort. We propose a new cohort based on the most updated versions of HAPUI criteria as determined by the CMS, NPIAP, and AHRQ guidelines [1, 2, 3,

7]. Stays of patients younger than 15 or resulting in death were excluded. More importantly, the cohort only considers new PUIs or PUIs which have gotten worse by discharge. Thus, it is necessary to determine the admission and discharge PUI stages for each stay. The admission stage is the first numerical staging which happens within 24 hours of admission. Deep tissue injury is coded as stage 4, and unable to stage is ignored. Since all stays must have an admission stage, if no staging meets the above criteria, the admission stage is set to 0. The discharge stage is set as the last numerical staging above 2, considering deep tissue injury as stage 3 and unable to stage as stage 5, which occurs 24 hours after admission. Unable to stage is set to stage 5, since regardless of admission stage, a PUI case is identified. Based on the NPIAP documentation [42], deep tissue injury is either a stage 3 or 4 PUI. Therefore to have the most certainty, we code deep tissue injury as stage 4 at admission and stage 3 at discharge. Only stays with discharge stage worse than admission stage are considered as PUI cases.

The properties of all three cohorts are shown in Table 3.1 and the overlap between the case stays for them is shown in Figure 3.2. Around 7% of positive samples are unique to our cohort, which separates it from the rest. Our cohort has the highest overlap with CANTRIP with over 21%. Only around 5% of HAPUI cases are shared among all three cohorts emphasizing the importance of cohort definition in HAPUI detection.

Table 3.1: Cohorts properties and composition.

Cohort	Exclude deceased? (Q1)	Min Age (Q2)	Cutoff period (Q3)	Uses all PUI stages codes (Q4)	Inc DTI or unstageable as +?(Q5)	Uses notes (Q6)	Only worsened or new PUIs?(Q7)	Uses ICD9 codes for+ (Q8)	Cutoff stage (Q9)	#of + (all) samples	%of + class
Golden	**Yes**	**15**	**24**	**Yes**	**Yes (conditional)**	**Yes**	**Yes**	**No**	**2**	**3012 (44859)**	**6.7 %**
CANTRIP [28]	No	15	48	Yes	Yes	Yes	No	No	1	4261 (50376)	8.4%
Cramer [22]	No	18	24	No	No	No	No	No	2	1572 (50276)	3.1%

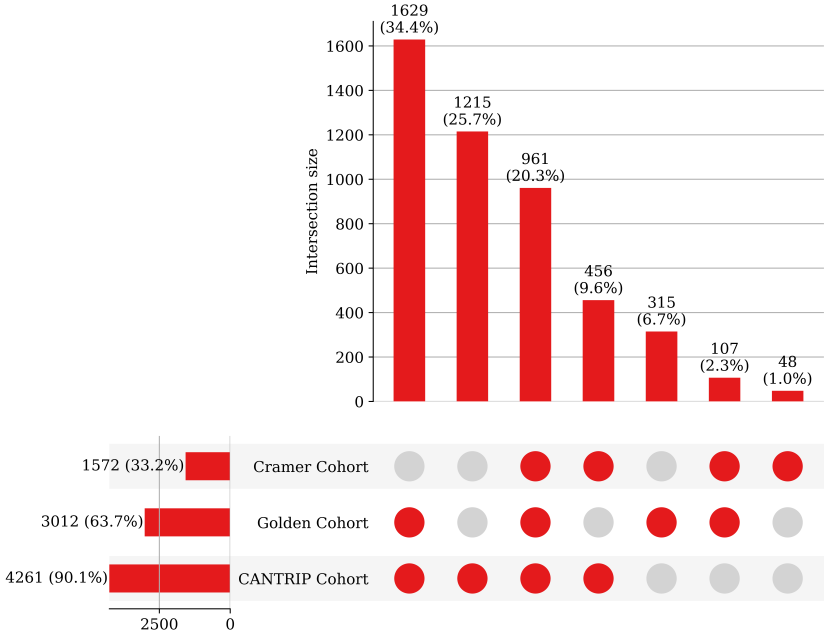


Figure 3.2: Overlap of Case stays across the three cohorts.

3.3 Experimental Settings for Cohort Comparison

Here we describe the details of the experiments we carried out to analyze the performance of different cohorts for HAPUI detection task.

3.3.1 Feature Construction

We construct hospital stays' features using a subset of their notes. For each hospital stay, we determine the feature cutoff time using two criteria. First, from the chartevents data, the time for the first staging occurring at least 24 hours after the admission time, regardless of its stage value, is extracted. Second, if any of the List 3 keywords appear in notes, the earliest time among them is considered. Then the feature cutoff point is the earliest time between these two times. If none of the two times exist, the features' cutoff point is determined as follows.

All the admissions not in this set (i.e. admissions containing PUI events of interest) are put into one class, with their notes' durations regarded as a random variable. We then fit a collection of distributions from scikit-learn package [16] and choose the most appropriate distribution for this duration of notes. We then sample from this estimated distribution for our admissions with no PUI event. To account for matching between this duration of notes and the actual full length of notes, we sorted both sampled note durations from the distribution and admissions themselves based on their actual note length and used the sorting index to assign limited durations for notes to admissions. If the assigned note duration is more than the full-length note, full-length notes is used for feature construction (i.e. the feature's cutoff point would be the last note's time).

Stays with an earlier admission than discharge time, no notes before the feature cutoff time, and admission notes where the main complaints had any variant of the List 3 keyword were removed from our study. After the feature's cutoff time is determined for each stay, all notes timed before that are aggregated into one document. This is to prevent labels from leaking into the features. As such, List 3 keywords will not appear in the constructed documents. Similarly, suppose the first staging is before the first keyword mention, or there is no mention in notes. In that case, since we anticipate broader PUI-indicative words appearing in notes after that staging time, we do not permit the concatenation of these later notes into the feature vector. The term frequency, inverse document frequency (TF-IDF) of this document is the feature vector of each stay for the tree-based classifier (considering a total of 5000 words in the vocabulary). For our sequential-based neural network, a sequence of 800 words of each document is used for training and prediction. All cohorts use the same features, as the features' cutoff point is independent of the cohorts' definitions, ensuring that all cohorts have access to the same amount of data, enabling us to fairly compare their performance.

3.3.2 Classifiers and Evaluation Metrics

We chose two classifiers to illustrate that the cohorts' relative performance is independent of the classifier choice, as long as it performs well enough. Gradient boosting, a tree-based classifier, and a sequential neural network-based classifier, consisting of input word embedding, global max-pooling layer, and several dense layers were selected because of their superior performances compared to the other classifiers tried (i.e., decision tree, logistic regression, support vector machine, multi-layer perceptron, random forest, and AdaBoost). For the experiments, data is split into 80% training and 20% test. 5-fold cross-validation is used to determine the best classifier hyperparameters. Finally, we report the test performance across ten test and training data partitions. Given the unbalanced class distributions, we report AUPRC (Area Under the Precision-Recall Curve) and AUROC (Area Under the Receiver Operating Characteristic Curve) metrics.

3.3.3 Defining Training and Test Sets in the Cohorts

Due to the different cohort criteria, the samples for each are different. Despite that, cohorts' samples have much overlap, as shown in Figure 3.2. For a fair comparison, we created two different test sets for all three cohorts. Firstly, we asked an expert nurse to manually label 85 admissions based on their notes and chart data. These 85 admissions were chosen from the conflicting samples whose labels are different across the three cohorts. Each constituent subset is proportional to the size of the conflicting samples of the two particular cohorts. Out of the 85 admissions, our expert marked 29 as positive for HAPUI.

Separately, once we obtained these expert annotations, we further augmented this small test set by considering the admission which the three cohorts all have consensus on. Considering the 29 HAPUI positive admissions and the 219 HAPUI positive admissions in the consensus set, to maintain the actual prevalence of around

7%, we added 3846 negative admissions from the consensus set. Sampling these 3846 negative consensus admissions were repeated ten times to obtain 10 different partitions of training and test data.

For each cohort, the training samples are then the samples not part of the chosen fixed test set, and the labels are determined by the respective cohort criteria for HAPUI. Therefore, even though the test samples are the same for the cohorts, each cohort has a different training set and training labels.

3.4 Results for Cohorts Comparison

In this section, we describe our experiments for comparison of cohorts and interpretation of classifiers.

3.4.1 Cohort Definition's Impact on Classifiers' Performance for HAPUI Detection

We summarize the performance of the gradient boosting, and neural network classifiers trained on labels determined by the three cohort definitions. We report the AUPRC and AUROC in each case in Table 3.2 and Table 3.3 for both test sets, ten combined test sets and the expertly annotated test sets, respectively. Our cohort definition outperforms the other two cohorts for both classifiers and both metrics. A one-sided paired t-test between our proposed golden cohort and the next best performing cohort definition (CANTRIP) showed a p-value of 0.0149 and 0.0245 for AUPRC and AUC of the better performing gradient boosting classifier and machine epsilon for the rest of the classifiers and cohorts. This proves the merits of the golden cohort definition.

Table 3.2: Classifiers' performance trained on the three PUI cohorts on the combined expert annotations and non-conflicting test set.

Cohort	GB Average AUPRC (SD) p-value	GB Average AUC (SD) p-value	NN Average AUPRC (SD) p-value	NN Average AUC (SD) p-value
Golden	**0.4908** (±**0.0126**)	**0.9045** (±**0.0021**)	**0.4714** (±**0.0180**)	**0.8943** (±**0.0041**)
CANTRIP[28]	0.4797 (±0.0182) [0.0937]	0.8991 (±0.0035) [4.31e-08]	0.4439 (±0.0189) [0.0005]	0.8912 (±0.0046) [0.0572]
Cramer[21]	0.3739 (±0.0237) [6.347 e-07]	0.8789 (±0.0044) [0.0004]	0.3941 (±0.0129) [2.399 e-07]	0.8665 (±0.0065) [3.831 e-06]

Table 3.3: Classifiers' performance trained on the three PUI cohorts on the expert annotations of 85 conflicting samples across the cohorts

Cohort	GB AUPRC	GB AUC	NN AUPRC	NN AUC
Golden	**0.4424**	**0.6071**	**0.4835**	**0.5714**
CANTRIP[28]	0.4329	0.5751	0.4192	0.5511
Cramer[21]	0.3858	0.5775	0.3727	0.5480

3.4.2 Significant Features for Classifiers and Cohorts

For each cohort, we analyzed the sensible words that were recognized as the most important words by the two classifiers. For gradient boosting, we directly used the feature importance of the classifier determined by the purity criteria. For the sequential neural network, we used the 'shap' [42] package by using the first 1000 samples of the training data to explain the first positive test sample. Since our focus is not on preprocessing data, some of the important words discovered were general terms like 'sex' or 'patient.' Table 3.4 illustrates the non-general important features generated. Our cohort extracted the most specific HAPUI terms among the three cohorts using both classifiers.

Table 3.4: Significant words for combinations of classifiers and cohorts (sorted by importance).

Cohort	Gradient Boosting	Neural Networks
Golden	**line**, **svc**, **vent**, **coarse**, **aspirin**, **suctioned**, **picc**, **lower**, **changes**, **disposition**, **coccyx**, **vanco**, **abgs**, **chamber**, **facility**, **tan**, **paralyzed**, **dialysis**, **osteomyelitis**, **wound**, **abscess**, **abnormalities**	**coherent**, **extended**, **motrin**, **qtc**, **syncope**, **bradycardia**, **plaque**, **fracture**, **pressure**, **valve**, **echo**, **ibuprofen**, **sat**, **reassess**, **ventricular**, **facility**
CANTRIP[28]	doppler, line, picc, resp, lower, remains, ed, allergies, pa, tan, dictated, svc, weaning, tracing, coccyx, suctioned, residuals, abnormalities, breath, myocardial, fascicular	ambulatory, sutures, delayed, consciousness, instructions, asa, clopidogrel, facility, chlorhexidine, bisacodyl, shower, syncope, fracture, prolapse, disposition, plavix, sodium, injury, hgb, spinal, qtc, trauma, sat, bradycardia, stenosis
Cramer[22]	doppler, abg, paracentesis, habitus, peep, lower, anasarca, sedation, anesthesia, debridement, cavity, respiratory, weaning, coarse, paralyzed, desaturation, ciprofloxacin, fent	shower, sah, walking, clopidogrel, chlorhexidine, stairs, procedural, gluconate, aid, coumadin, hgb, consciousness, systolic, bisacodyl, central, perfusion, bid, veins, rehab, instructed, fluids, ultrasound, extremity, femoral, mid, syndrome

3.5 Cohort Definition for HAPUI Detection, Next steps and Reflections

We defined a new standardized HAPUI cohort criterion using EHR data that considers all available data sources, including text, chartevents and procedures, and diagnosis codes, unlike the current cohort definitions. Our cohort definition is also based on clinical and regulatory guidelines, which have not been fully incorporated into the previous cohort definitions. We validated the usefulness and accuracy of our cohort definition using two carefully designed test sets of fully manually labeled set, and a larger hybrid set of fully manually labeled set and samples with label agreements across cohorts, by showing its superiority on both AUPRC and AUROC while using two kinds of classifiers, neural network and gradient boosting.

Perfect abidance by the CMS guideline of only considering worsening and non-healed PUIs requires matching of admission and discharge PUIs. A patient may be admitted with more than one PUI and discharged with more or fewer ones. The worsening condition should be checked for each distinct PUI individually. However, given limited data in the chartevents table in MIMIC-III, our cohort criteria assume all stays are associated with one PUI.

Yet even if the perfect criteria for HAPUI is achieved, depending on the construction of the cohorts, we still need to deal with the uncertainty of the sources and subjectivity of guidelines; therefore, leveraging a computational framework that explicitly deals with uncertainty would be extremely helpful. This lays out the foundations for the next piece in this thesis, uncertainty-aware sample-specific perturbations Co-teaching based classifier.

Chapter 4

CrowdTeacher: Robust Co-teaching with Noisy Sparse Answers and Sample-specific Perturbations

A standard criterion for HAPUI helps with benchmarking it but, unfortunately, will not eradicate the uncertainty of the sources. Thus, developing a computational framework that is designed to leverage the uncertainty of samples is truly advantageous. The intuitive idea is to transfer the uncertainty of labels into the features used for learning. In this project, we adapt some of the recently proposed algorithms for noisy labeling with newly developed crowdsourcing with uncertain labels methods and attempt to bridge these methods for more robust classification.

4.1 Crowdsourcing: Toward Labeling Unlabelled Data

Labeled data is critical for the success of increasingly more complex classifiers. Unfortunately, getting access to large quantities of high-quality labels can be cost-prohibitive in many applications. For example, in the medical domains, it may take a clinician a long time to annotate the health records of thousands of patients. One alternative to this cumbersome process is to gather labels using crowdsourcing, where remotely located workers are utilized to perform the task of labeling the data. Although these crowdworkers individually may not be as accurate as an expert, constructing the true label from their aggregated opinions can approximate the accuracy of an expert. However, the subjectivity of annotators and their different level of expertise adds noise to the labeling process. To model this noise, most studies either focus on modeling the reliability of annotators and their correlations to reflect it in the label aggregation phase or coupling classifier training with learning the annotators' trust parameters. Yet, learning through crowdsourcing-based models may still fail in the presence of differing annotations, and unreliable annotators [64].

A promising approach for dealing with noisy labels within complex classifiers is Co-teaching [31]. Under the Co-teaching paradigm, two peer neural networks are trained separately, and particular samples are exchanged between the networks to decrease the error of the two models and obtain a more accurate model. Co-teaching methods have shown great promise for computer vision problems with noisy labels. Co-teaching can naturally counteract crowdsourcing noise since it filters out noisy samples initially and only adds them at later training stages when they will be of greater value. However, Co-teaching applies the same weight to each sample. This can result in the classifier incorrectly learning from samples because they either have fewer annotations or diverging human labels.

To accommodate this limitation, we propose to leverage the certainty of samples from the label aggregation phase to inform the filtering process of Co-teaching, which has not been investigated before. Our model, CrowdTeacher, uses a perturbation scheme based on the samples' uncertainty to improve the robustness of the Co-teaching paradigm. Given the availability of samples' uncertainty from the label aggregation step, our model takes advantage of this information to counter the inherent noise through perturbation of the input space. In addition, the framework gives more priority to the more confident samples of the classifier during the learning process. Thus, we tackle the problem of classification with features and crowdsourcing labels using three mechanisms:

- Estimation of the features' distributions to generate synthetic data, which is then used to perturb each sample in an additive manner, proportional to its estimated label's uncertainty.
- Enhancement of Co-teaching by knowledge distillation, i.e., a student-teacher model of a simple and a complex network to accommodate smaller tabular data.
- Utilization of the perturbed samples as input to the above classifier to further differentiate uncertain and certain training points based on their loss in each epoch

4.1.1 Notations for Learning with uncertain Crowdsourcing Labels

We refer the reader to Definition 2.2.1 for defining the classification with crowdsourcing annotations problem (in Section 2.2). Here we introduce the notations we will use in the rest of this chapter to tackle this problem.

Table 4.1: Summary of Notations for CrowdTeacher.

Symbol	Description
N	Number of Samples
R	Number of Annotators
K	Number of Classes
α	Perturbation Fraction
$\mathbf{X}_{tr}$	Training feature matrix
$\mathbf{A}$	Answer matrix of all annotators
$\mathbf{S}$	Synthetic feature matrix
$\widetilde{\mathbf{X}_{tr}}$	Perturbed training samples feature matrix
F_c	Set of continuous features
F_d	Set of all discrete features
$\mathbf{P}$	Class probability matrix
c_i	Certainty of i-th

4.2 Uncertainty-aware Perturbation Scheme and Modifying Co-Teaching

Our idea is to enhance the Co-teaching framework in Section 2.4 to account for the uncertainty associated with the estimated truth label of the sample. We posit that employing uncertainty-aware perturbations will yield more robust classifiers since these perturbations can potentially counter the noisiness of the labels by introducing explicit, directed, and intentional randomness in the input feature space proportional to the level of uncertainty in samples' estimated labels, inspired by works that show the equivalency of perturbations in labels, input and network structure space [10].

We introduce a perturbation-based scheme to the Co-teaching framework so the trained model will be more robust to sparsity and unreliability in the annotations and potential incorrect annotations. For each mini-batch update of Co-teaching, synthetic samples are generated and used in perturbation of each sample *dependent* on the certainty of the estimated truth label. Thus a sample that has more certainty in its label will be perturbed more, whereas a sample with fewer annotations is likely to have less perturbation. The perturbed samples are subsequently used to train the

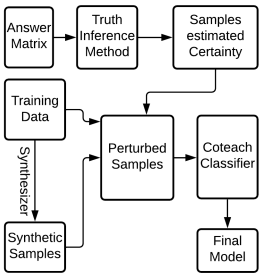
classifier. The flowchart of CrowdTeacher showing its different components is shown in Figure 4.1a.

4.2.1 Generating Synthetic Samples

To improve the robustness of the Co-teaching framework, CrowdTeacher generates synthetic samples of the data, previously introduced in Section 2.3, which are then used to perturb the samples for classifier's training. Since in many real-world problems, features are intricately correlated, to preserve this correlation structure, synthetic sample generation is often a more realistic approach than perturbing each feature separately. Any data synthesizer with reasonable data generation performance may be used. For the purpose of our project, we selected three representative data synthesizers: Conditional GAN (CTGAN) [74], TVAE [74], and Gaussian copula [54]. CTGAN can handle mixed feature types (discrete and continuous) and has been shown to perform competitively with other GAN-based, VAE-based, and Bayesian network-based data synthesizers for vision benchmark datasets [54]. It is noteworthy to mention that the data synthesizer is not tied to the learning task and can be used as a stand-alone tool.

To generate synthetic data within CrowdTeacher, the training feature matrix $\mathbf{X}_{tr}$ is fed to the synthesizer. For the CTGAN synthesizer, the discrete features F_d are explicitly specified as they are modeled differently from the continuous features F_c. Once the synthesizer has estimated the data distribution, any number of samples can be drawn. For CrowdTeacher, we generate the synthetic set $\mathbf{S} \in \mathbb{R}^{N \times M}$ with N synthetic samples once and assume each synthetic sample is a unique perturbation source. Although $\mathbf{S}$ is drawn once and is the same size as our training data to minimize the computational footprint of our model, the synthetic set can potentially be re-drawn at each mini-batch of the Co-teaching framework for a larger population.

(a) End to End flowchart of CrowdTeacher



(b) Uncertainty-aware sample-specific perturbations

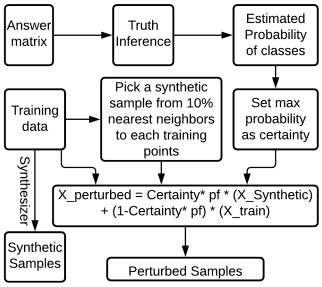


Figure 4.1: Conceptual Framework for Uncertainty-aware sample-specific perturbations and CrowdTeacher.

4.2.2 Sample-specific Perturbations

The generated synthetic samples, $\mathbf{S}$, fail to account for the uncertainty of the estimated sample label as the synthetic samples are only dependent on the initial training data. Thus, we introduce a mechanism to take advantage of the uncertainty that arises from the truth inference method to individually perturb each sample. For the purpose of illustration and experimentation, we focus on the D&S algorithm [23], but note that CrowdTeacher can be used with any robust truth inference method that quantifies the label uncertainty for each sample. The D&S algorithm (defined in Section 2.1.2) takes as an input the matrix of annotations ($\mathbf{A}$) and models annotators by a confusion matrix 2.1.2, in addition to the class priors. D&S outputs a matrix $\mathbf{P} \in \mathbb{R}^{N \times K}$, where the P_{ik} element shows the probability that sample i is of class k. The certainty of each sample, c_i, is then determined to be the maximum probability across all the classes:

$$c_i = \max_{k \in K} (P_{ik}) \quad \forall i \in N \tag{4.1}$$

Choosing an appropriate simulated sample for perturbation. Since the data synthesizer may generate synthetic samples that are quite different from the original dataset and may lead to more uncertainty with respect to the truth label, we use

k-nearest neighbors (KNN) to identify reasonable close samples from **S**. For each sample, KNN algorithm is run to find the top 10% closely simulated samples to it. A simulated data point, s_i, is then randomly chosen from this top 10% and used in the CrowdTeacher process to perturb the original point.

Perturbation. Each sample x_i is perturbed using the simulated data point s_i based on the uncertainty, c_i and a user-specified perturbation fraction $\alpha \in [0, 1]$ to generate the perturbed sample $\widetilde{x}_i$. Let s_{ij} represent the j^{th} feature of sample s_i. If the j^{th} feature is continuous, the value for the synthetic, perturbed sample $\widetilde{x}_{ij}$ is a convex combination of the original and simulated sample:

$$\widetilde{x}_{ij} = (1 - \alpha c_i)x_{ij} + (\alpha c_i)s_{ij}, \quad \forall i \in N, \quad \forall j \in F_c \tag{4.2}$$

For the discrete features, we use c_i and α to calculate the number of discrete features to swap. Let $|F_d|$ denote the number of discrete features in the dataset, then the number of discrete features to swap for each sample x_i, f_d^i is calculated as:

$$f_d^i = round(\alpha c_i |F_d|) \tag{4.3}$$

Then f_d^i features are randomly selected for perturbation from the original discrete feature set and denoted as $F_{d_p}^i$. For each feature, j in this perturbation set, the feature values are replaced with the synthetic sample value s_{ij}.

$$\widetilde{x}_{ij} = s_{ij}, \quad \forall i \in N, \quad \forall j \in F_{d_p}^i \tag{4.4}$$

The complete procedure for generating uncertainty aware sample-specific perturbations is depicted in Figure 4.1b.

4.2.3 Knowledge-Distillation-based Co-teaching for Smaller Tabular Data

To combat the large performance variations associated with running the Co-teaching algorithm on smaller-sized tabular data, we incorporated the student-teacher idea from knowledge distillation [32]. In essence, instead of two peer networks with the same architecture, we used one simple and one complex network in the classifier such that the number of hidden units for the simpler network is half of the more complex one. Also, the variance of the initial weights for the simpler network is twice that of the more complex one. Our experimental results proved these modifications to be helpful for both the convergence of the two networks in achieving more similar evaluation metrics and overall better performance across different synthetic datasets.

Algorithm 1 provides the pseudo-code for CrowdTeacher.

4.3 CrowdTeacher Experimental Settings

Here we describe the baseline methods for comparison, introduce the datasets and the describe annotation generation process.

4.3.1 Baseline Methods

The best-performing methods from crowdsourcing studies (see Section 2.2) and incremental variants for ablation study of CrowdTeacher are chosen as comparison models. That is, the original Co-teaching algorithm and Co-teaching using only uniformly perturbed input are also compared to illustrate the advantage of certainty-aware perturbation and perturbation in general. All methods share the same base classifier, a neural network with one hidden layer of $\frac{|F_c|+|F_d|}{4}$ units. Sequential methods discussed in Section 2.2.1 all employ the same truth inference method (D&S) and are distinguished with * below.

Algorithm 1: CrowdTeacher.

Input: Training Features $\mathbf{X}_{tr}$, Answer matrix $\mathbf{A}$, Perturbation Fraction α
Output: $Model$
Train synthesizer to generate synthetic data:
$Data\_sampler \leftarrow Synthesizer(\mathbf{X}_{tr})$
Generate N samples from resulting sampler: $\mathbf{S} \leftarrow Data\_sampler(N)$
Run truth inference method to get class probabilities:
$\mathbf{P} \leftarrow D\&S\_Algorithm(\mathbf{A})$
/* Generate perturbed samples $\widetilde{\mathbf{X}_{tr}}$ */
for $i = 1, \cdots, N$ **do**
    Set sample's certainty using Eq. (4.1)
    Sample s_i from 10% closest samples of synthetic samples $\mathbf{S}$ to x_i using KNN
    /* Generate continuous features */
    for $j \in F_c$ **do**
        Generate feature $\widetilde{x}_{ij}$ according to Eq. (4.2)
    end
    /*Generate discrete features*/
    Calculate f_d^i using Eq. (4.3)
    Sample discrete features to perturb: $F_{d_p}^i$ from F_d such that $|F_{d_p}^i| = f_d^i$
    for $j \in F_{d_p}^i$ **do**
        Generate single feature value $\widetilde{x}_{ij}$ according to Eq. (4.4)
    end
end
Train Co-teaching Algorithm on Perturbed Samples:
$Model \leftarrow Co\_teaching(\widetilde{\mathbf{X}_{tr}})$

- Naive baseline*(Base_clf) [23]: Base classifier trained with D&S labels.
- Simultaneous Expectation-Maximization (S-EM) [9]: An algorithm that jointly learns the classifier and annotators' parameters using the EM algorithm.
- Dr. Net [29]: An individual annotation based model that separately learns each annotator's labels and their weights.
- CrowdLayer (CL_MW and CL_VW) [59]: An algorithm that estimates ground truth first and replicates each annotator's labels via a simple final layer. This final layer is removed at test time. The number of parameters for the last layer determines the CrowdLayer variant. We evaluated the vector of weights (VW)

and matrix of weights (MW) variants.

- Vanilla Co-teaching*(V_Coteach) [31]: The original Co-teaching algorithm trained with D&S labels.
- Co-teaching with uniform perturbation*(P_Coteach): The Co-teaching algorithm trained on D&S labels and synthetic samples.
- CrowdTeacher*: Our proposed method with the Co-teaching algorithm trained on D&S labels and sample-specific certainty-informed perturbed samples.

As S-EM and Dr. Net constantly performed poorly compared to the other methods, we removed them from the plots for better readability.

4.3.2 Annotation Simulation

For our evaluations, we fix the number of annotators to be 5 ($R = 5$). To simulate the annotators' behavior, we use two parameters: (1) mean reliability or the average chance of the annotators labeling a positive sample correctly and (2) variability in annotators' expertise or the difference in their competency. We set the distribution of samples having 1 to 5 labels as $[\tau\ , 0.55(1-\tau), 0.27(1-\tau), 0.13(1-\tau), 0.05(1-\tau)]$ and vary the parameter τ for our experiments. Note that τ also dictates the average number of labels per sample. Conventionally in crowdsourcing, to generate each annotator's reliability, the Beta distribution is used. After specifying each annotator's reliability from the previous step, its labels are produced by randomly selecting (100-reliability) percent of positive cases and switching their labels into negative 0. Flipping negative samples to positives occurs at 0.01 times this rate. Samples not assigned to specific annotators are designated with -1 in the answer matrix ($\mathbf{A}$).

For HAPUI detection task, based on nursing research [63] we set the mean reliability of annotators to 77%. Similarly, for the length of stay prediction, we use these

research works that estimate the accuracy of physicians in predicting the length of stay of patients at admission [30, 47] and set their mean of reliability at 77% percent, assuming more senior physicians.

4.3.3 Datasets

Synthetic Datasets. To analyze the performance of our framework on a non-specific dataset for which we already know the ground truth, we generated synthetic data that mimics real-world features and a range of annotator reliabilities.

Statistical distribution families: To mimic real-world features, each set of features, which resembles correlated features encountered in practice, families of continuous and discrete distributions were used to generate the synthetic data. Specifically, we used Normal, Beta, Wald, Laplace, Binomial, Multinomial, Geometric and Poisson distributions. The parameters of the distribution for a feature within each family are randomly selected from a specified range. 5 features were chosen from each family to have a total of 40 features.

Output: The ground truth labels are assigned based on a polynomial combination of feature values. Each feature's coefficient value is chosen randomly. The exponent for each feature is also randomly chosen in the range [1,4]. To assign labels and model class balance (% of positive samples), outputs falling in percentiles below the level of balancedness are assigned to the positive class.

Noise level: Two versions of labels are created. Labels for a specified percentage of samples are flipped to obtain the noisy truth used for annotation generation. However, the true labels before flipping are the ones used for evaluation purposes. This mimics the availability of noisy labels in practice.

HAPUI Dataset. We included hospital stays of individuals over 20 years old with the length of stays between 2 days and 120 days. A hospital stay was considered positive if there was a presence of the ICD-9 diagnosis code associated with pressure

ulcers, and there was a mention of PUI in the notes. A hospital stay was negative if there was no indication of PUI in both the ICD-9 codes or the notes. 10518 samples were included in the total, with 31% of them belonging to the positive class. Features used are the demographics, the number of ICD-9 diagnosis codes (except PUI-related codes), and the average of lab measurements during the first and second day of admission.

Length of Stay Dataset.

For this task, we randomly sampled 5000 hospital stays from the MIMIC-III dataset. We set the class for by assigning hospital stays with a duration greater than 7 days as class 1, and class 0 otherwise, similar to [70]. The ratio of positive samples in this task is 28%. Given that we have some uncertain annotations regarding patients length of stay from admitting physicians in the beginning, we focus on using only their lab measurements on the first day and also add the number of ICD-9 diagnosis codes to see if we can classify new patients' stay as short (less than 7 days) or long (more than 7 days).

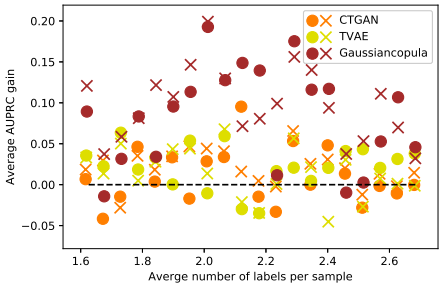
4.4 CrowdTeacher Results

Since the datasets are imbalanced, we evaluate all the models based on AUPRC. AUPRC offers a holistic picture of CrowdTeacher's predictive performance, independent of the classification threshold choice. We partition each dataset into 80% training & 20% test.

4.4.1 Synthetic Dataset

Sensitivity to the choice of synthesizer. To reason about the effect of using different synthesizers on CrowdTeacher performance, we compared the average gain gleaned from using CrowdTeacher with CTGAN, TVAE, and Gaussian copula synthe-

(a) Different synthesizers.



(b) Perturbation fraction.

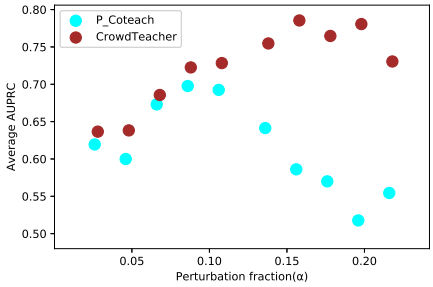


Figure 4.2: CrowdTeacher Sensitivity to perturbation fraction and synthesizer choice (in Figure 4.2a circles/crosses show gain w.r.t. P_Coteach/V_Coteach accordingly).

sizers compared to using the next two top-performing baseline methods of P_Coteach and V_Coteach, respectively shown by circle and cross markers in Figure 4.2a. Firstly, we can observe that the Gaussian copula has the largest gain among the three synthesizers. However, employing the two other synthesizers for CrowdTeacher would still be advantageous in terms of predictive performance for many of the sparsity settings. Since the Gaussian copula synthesizer performed well, we use the Gaussian copula for all the experiments after this.

Sensitivity to perturbation fraction (α). To better analyze the effect of the perturbation fraction, α, we varied it between [0.01, 0.2] and evaluated the performance of CrowdTeacher and P_Coteach (the two perturbation-based algorithms). Figure 4.2b depicts the average AUPRC of P_Coteach and CrowdTeacher as α increases with the average number of labels set to 2.34. It can be seen that CrowdTeacher constantly outperforms P_Coteach regardless of the chosen perturbation fraction, signaling its robustness. From the results, we can observe that there is an optimal range of α for achieving the greatest benefit from CrowdTeacher and that both a very low ($\alpha \leq 0.05$) and a very high ($\alpha \geq 0.2$) perturbation fraction reduces the usefulness of CrowdTeacher but does not diminish it. Given this trend, the remainder of our experiments uses $\alpha = 0.11$.

Predictive Performance. Figure 4.3a shows the performance of baseline crowd-

sourcing and Co-teaching variants against CrowdTeacher for various density settings on the synthetic dataset. Matching our intuition, all methods experience an increase in AUPRC as the average number of labels per sample increases, which exposes methods to less noisy annotation. All Co-teaching based methods (CrowdTeacher, V_Coteach, and P_Coteach) always outperform both crowdlayer variants and also Dr.Net and S-EM. The last two always performed the worst and therefore were excluded from these plots. Even though the base classifier performance improves with more labels, its performance gap with Co-teaching based methods still remains large in all densities. Across a wide range of label densities, using CrowdTeacher leads to a significant boost in AUPRC, compared to the other two Co-teaching based methods, even with as low as only 1.68 labels per sample. Furthermore, we observe that V_Coteach performs worse than P_Coteach in very sparse settings (average number of labels < 2.1), but as the number of labels increases, it catches up with P_Coteach and even outperforms it at higher densities. Another interesting observation is that beyond an average of 2 labels per sample, all three methods reach a plateau of performance and only improve negligibly as a function of the increased number of labels.

(a) Synthetic dataset. (b) PUI dataset.

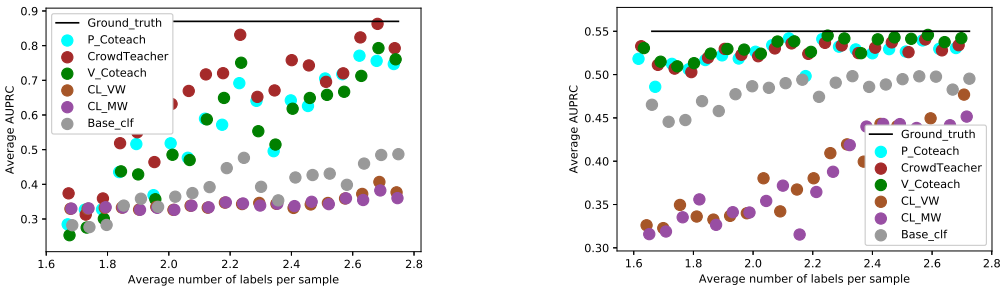


Figure 4.3: CrowdTeacher performance on Synthetic and PUI data as average number of labels per sample increases, averaged on 10 and 4 initializations respectively.

Smoothing Perturbation with Coteach. We compared the impact of the perturbations achieved through synthetic samples on our method CrowdTeacher and a simpler perturbation scheme we denote as Smoothing_Coteach, where we perturb

each feature independently. For continuous features, we use the feature value of that sample itself to either increase or decrease it weighted by its certainty and perturbation fraction. For discrete features, we first determine the number of discrete features that should be perturbed according to the sample's certainty and perturbation fraction. We then choose a discrete value among the valid choices for each chosen feature (excluding the feature value of the sample itself) based on its estimated prevalence randomly. Here the D_j denotes the estimated multinomial distribution for feature j. The smoothing perturbations can be described using Equations (4.5), (4.6), and (4.7). Here D_j shows the approximation for the multinomial distribution of discrete feature j.

$$\widetilde{x}_{ij} = (1 - \alpha c_i)x_{ij} + (\alpha c_i)x_{ij}, \quad \forall i \in N, \quad \forall j \in F_c \tag{4.5}$$

$$f_d^i = round(\alpha c_i |F_d|) \tag{4.6}$$

$$\widetilde{x}_{ij} = x_{\hat{i}j}, \quad \forall i \in N, \quad \forall j \in F_{d_p}^i, \text{ such that } i \neq \hat{i} \text{ and } x_{\hat{i}j} \sim D_j \tag{4.7}$$

We generated 3 synthetic datasets of varying complexities with regard to their features and their correlations. Our experiments confirm the significance of synthesizer in CrowdTeacher performance. As shown in Figures 4.4, 4.5 and 4.6, depending on how correlated the features are, Smoothing_Coteach performs worse than CrowdTeacher across different sparsity settings, especially in lower sparsities. Since in many practical settings, features can be quite correlated, using a synthesizer instead of smoothing perturbation will result in higher predictive power.

4.4.2 PUI Dataset

To assess CrowdTeacher's performance on real data, we evaluated it on the bedsore detection task with 10k samples. Figure 4.3b shows how the performance of the selected

Figure 4.4: Smoothing_Coteach v.s. CrowdTeacher-Correlated Features.

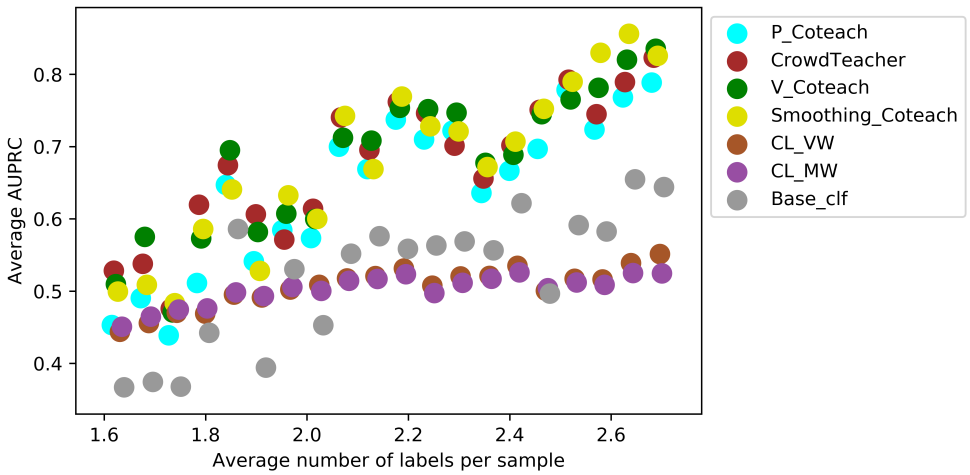


Figure 4.5: Smoothing_Coteach v.s. CrowdTeacher-Loosely correlated Features.

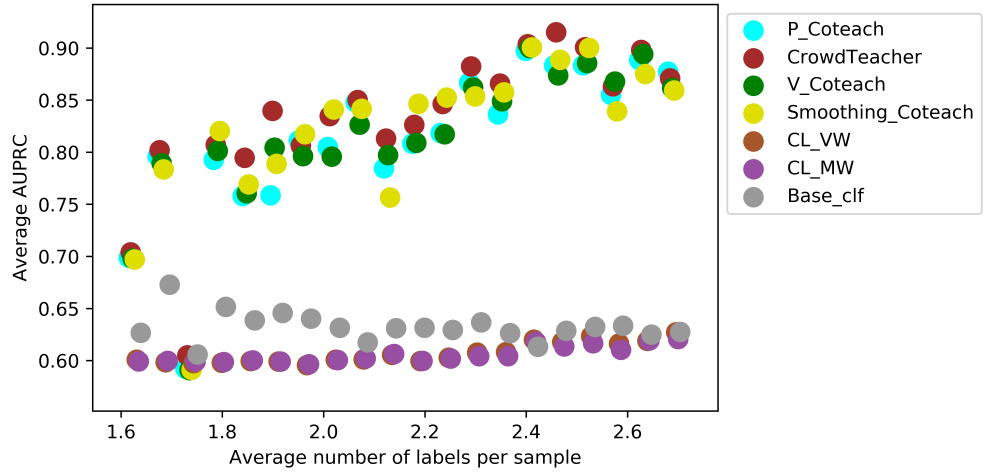
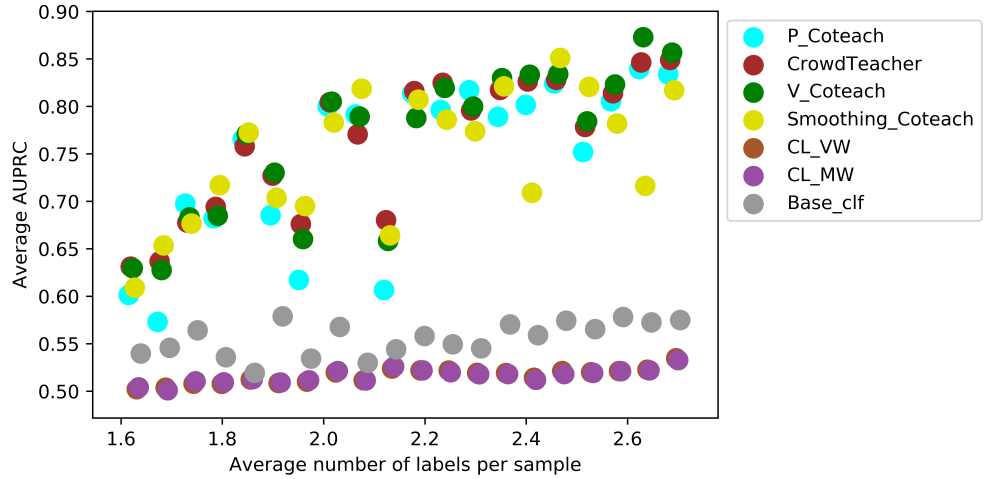


Figure 4.6: Smoothing_Coteach v.s. CrowdTeacher-Very Loosely correlated Features.



methods changes as the average number of labels per sample increases. We observed similar trends to the synthetic dataset here, too, in terms of Co-teaching variants' overall predictive benefit over other methods; however, the gap between Co-teaching variants and other methods is less noticeable. The range of AUPRC for all models on this dataset proves that this is a tougher learning problem, yet CrowdTeacher is able to outperform both P_Coteach and V_Coteach at multiple points, especially at lower densities, which are actually more practical for obtaining labels for hospital-acquired bedsores, while at other sparsity points it has comparable performance to these methods.

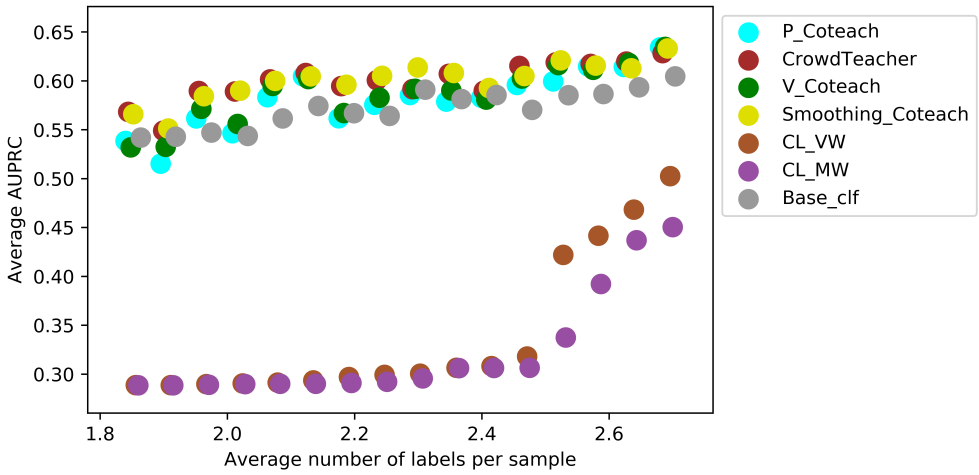
4.4.3 Length of Stay Dataset

To validate the performance of CrowdTeacher on another real-world dataset, we analyzed its performance across different sparsity settings for the length of stay prediction task. In Figure 4.7, we vary the average number of labels per sample from 1.85 to 2.85. The vertical axis shows the average test AUPRC of the methods across 20 runs. The perturbation-based variants of Smoothing_Coteach and CrowdTeacher have better performance than the other baseline methods, with CrowdTeacher often providing more boost than Smoothing_Coteach, especially with sparser annotations. Similar to synthetic and HAPUI datasets, V_Coteach and P_Coteach often perform better than the base classifier, demonstrating that incorporating uncertainty in learning through Co-teaching alone or Co-teaching and uniform perturbation boosts performance although less than the CrowdTeacher enhancements.

4.5 CrowdTeacher and HAPUI Detection

We proposed CrowdTeacher, a novel Co-teaching based approach that leverages certainty of samples from truth inference algorithms to apply sample-specific perturba-

Figure 4.7: CrowdTeacher performance for LOS prediction task across 20 seeds.



tions on training points and combines it with the Co-teaching paradigm to further filter noisy annotations and blend that knowledge in the training process. Our proposed approach bridges overarching themes and ideas from data augmentation, crowdsourcing, and learning with noisy labels and is agnostic to the truth inference method and the synthesizer used. To illustrate the predictive benefits of CrowdTeacher over similar methods, we conducted experiments on both synthetic and real datasets of different scales, and our results for both tasks (including for two real-world medical classification tasks) confirmed CrowdTeacher's performance edge for learning with crowdsourced labels.

Reflecting upon the consistently better performance of CrowdTeacher across densities for the synthetic dataset and its marginally better performance for the HAPUI task, we realized that using only the structural features for this task may have contributed to this poor performance. From Section 3, we observed how using unstructured features like notes considerably boosted the performance of all cohorts. Based on that, our next logical step for HAPUI detection is employing unstructured data for constructing features and developing an interpretable model that would help us get deeper insights into the most useful features.

Chapter 5

Discovering Better Features and Improving Performance Using Unstructured Notes

We've already seen that searching for the existence of HAPUI significantly increased the number of HAPUI cases for our analysis. Also, from Section 3 we observed that using text data, we were able to harness more predictive power for classifiers trained on all cohort definitions. In this section, we focus on textual features and a new preprocessing step to get a better understanding of the reasons behind these improvements, which would ultimately lead to better feature constructions benefiting the CrowdTeacher paradigm.

5.1 Text features, Missing Piece For HAPUI Detection

Many of the works using EHR for HAPUI focus on structured and static data. One Korean risk assessment study [19] developed a decision support tool based on a

Bayesian network risk model, resulting in a significant incidence reduction from 21% to 4%. This finding supported the usefulness of PUI risk alert tools. Unfortunately, this tool also similarly only used structured EHR data such as billing codes, which cannot provide real-time and accurate detection for PUI. Moreover, the model performance could be further improved by including unstructured data such as nursing notes, which contain useful patient information.

In practice, responsibilities for PUI prevention and treatment fall to the nurse. Yet nurse-collected data is not actively mined for valuable patient information. Nursing notes are one of the common unstructured data that includes the changes of patient vital signs, symptoms, and care may thus be more valuable and informative in PUI prediction and detection than the structured data. However, due to the high workload and insufficient communication between healthcare professionals, compared to structured data, unstructured data are very unlikely to be routinely used for decision-making. For example, the National Pressure Ulcer Adversary Panel developed a template with the needed documents to facilitate the discovery of severe PUI development through a review of the timeline of events [48]. In this 18-page general template, most PUI risk factors could only be captured by unstructured data such as skin outlook descriptions, re-positioning, and support surface as documented in nursing notes.

Therefore, it would be beneficial if PUI-related unstructured data, especially nursing notes, could be leveraged for PUI prediction prior to its occurrence for early detection or to inform nurses to implement appropriate interventions in time. Our proposed approach attempts to address some of these challenges through setting up a PUI detection pipeline that takes advantage of hospital notes and a negation-aware processing step before feeding it to a classifier for detection of PUI.

5.2 Dataset and Labeling Details

Here we describe the dataset we used and details of our cohort selection.

MIMIC-III Dataset: To ensure the replicability of our experiments and results, we used the openly available MIMIC-III dataset [34]. This dataset contains information of patients admitted to intensive care units (ICU) of a populated tertiary care hospital from 2001 to 2012. There are 49,785 unique hospital admissions for patients aged 16 years and older. These records come from 38,597 unique adult individuals [34].

Cohort selection: We chose hospital stays as the unit of analysis in this problem to reflect the real-world assessment of all the charts for a stay by nurses. After removing hospital stays with illogical attributes, e.g. those with a negative length of stay, about 50k unique stays remained. Since in MIMIC-III comparatively very few positive cases of PUI were present in the younger population, hospital-stays of individuals 20 years and younger were removed. We restricted our analysis to stays longer than 2 days and shorter than 120 days, since shorter stays provide insufficient notes for satisfactory representation, and extremely long stays are most probably erroneous records in this specific dataset. This further cuts down the number of unique hospital stays to about 26K. The available notes for each stay, therefore, will be its features.

Establishing Presence of PUI: For the purpose of deriving the most indicative HAPUI words, here we employ a more certain cohort than the cohorts defined in Section 3. Two sources of information for each hospital-stay, ICD-9 diagnosis codes, and notes, are used to determine the presence or absence of PUI. A hospital-stay is indicative of PUI from an ICD-9 perspective if any of the PUI ICD-9 codes in Definition 3.2.2 are found in its diagnosis codes. Similarly if any of the PUI keywords in Definition 5.2.1 or string versions of ICD-9 codes in Definition 3.2.2 are found in the notes, that stay is indicative of PUI from a notes perspective. Hospital-stays that indicate PUI in both sources constitute our positive class, and those with no indication of PUI in either will be labeled as negative. To avoid ambiguity in our

dataset, we discard stays that indicate PUI only in notes or only in ICD-9 codes. Note that ICD-9 codes at discharge time are only used for establishing labels and are never used as features in the prediction.

Definition 5.2.1 (PUI Explicit Keywords)**.** [Pressure Ulcer Prevention, Skin Surveillance, Decubitus Ulcers, Impaired Tissue Integrity, Impaird Skin Integrtiy, Bed Sores, Pressure Ulcer, Pressure sore]

Determination of PUI Case/Control Samples

Given that the ratio of positive samples (PUI) to negative ones (no PUI) is very low (3.5%), heavily inhibiting the learning algorithm's ability in distinguishing the two, we decide to apply a case-control design as a solution. A given positive sample is matched with 4 negative samples (stays), closest to it in terms of age, gender, the total length of stay, and ICU length of stay. To accomplish this matching, a 4 nearest neighbor algorithm was trained with all the negative samples, and then for each positive stay, the 4 closest negative samples without PUI were added to the pool. Negative samples do not have to be unique for each positive sample, and some negative samples might be matched with multiple positive ones, i.e, the selection process happened with replacement.

In total, 856 stays were marked as positive for PUI and using the case-control study, 2733 negative stays for PUI were selected for our experiments. Our final cohort consists of 3589 hospital samples, with a 31.3% positive to negative sample ratio.

Table 5.1: Properties of different Stages of Cohort Selection in MIMIC-III Dataset.

Cohort	Total # of Unique stays	# of PUI stays	# of no PUI stays
MIMIC-III total hospital-stays	50,027	1,249 (2.4%)	46,227 (92%)
Target age and length of stay cases	26,838	856 (3.1%)	23,886 (89%)
Case controlled stays	**3,589**	**856 (23.8%)**	**2,733 (76.2%)**

5.3 Data Analysis

Medical documents usually contain terms that only when considered in the context of a sentence can be interpreted as a symptom for the presence/absence of a condition [18]. Therefore, we propose a negation-aware processing step on hospital-stays notes. Our process identifies the negative mention of conditions according to the sentence context, putting a negation prefix right before its mention in the processed notes (e.g. no_edema) to create a distinguished word. After the negation, we transform each hospital stay's nursing notes (both before and after negation-aware processing step) into a vectorized feature representation. We explored three different classifiers to answer the following two questions:

- What is the effect of our proposed negation-aware framework on the performance of PUI detection?
- What are the most significant text features, and do they overlap with known medical factors of PUI?

Figure 5.1 provides an overall illustration of our proposed methodology.

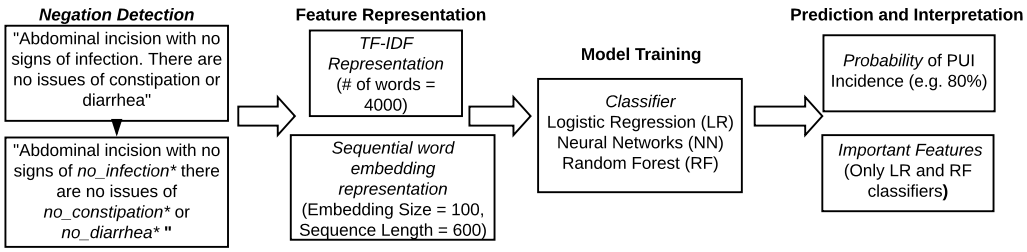


Figure 5.1: Overview of PUI Detection: Negation Detection, Model Training, Interpretation.

5.3.1 Negation Detection in Text Data

It is common for healthcare professionals to describe the absence of specific patient findings in clinical notes. For example, the sentence "showed no evidence of congestive heart failure or pneumonia" rejects the existence of congestive heart failure and pneumonia. Unfortunately, standard text processing does not handle negations. As a result, in many natural language processing pipelines, these can easily be mistaken for the presence of the conditions.

Before applying our negation detection scheme, notes were cleaned syntactically. All punctuation signs are taken out except for full stop and colon, which are used as a boundary for sentence segmentation. Standard stop-words that do not indicate the existence of a negation tone (words such as not and doesn't) are removed. We also remove all kinds of tab characters, end of line carriage return characters, consecutive multiple white space characters, and all special characters used to anonymize personal data such as [****]. Finally, all the characters in the text are lower-cased. To determine whether diseases or symptoms mentioned in the clinical notes were negated by the dictating physician, we combined two clinical text processing methods, Scispacy [52] and NegEX [18]. Scispacy extracts all mentions of named entities, including disease, medication, symptoms, and chemicals. NegEX uses defined regular expressions that cover several phrases indicating negation, filters out sentences with phrases that falsely appear to be negation phrases, and limits the scope of the negation phrase. First, our text processing determines the sentence boundaries in each text using the full stop and colon as the boundary to limit the scope of the negation. The mentions of named entities in each sentence are identified using Scispacy. NegEx is then run on the sentence to determine which named entities should be negated. As an example of our preprocessing step, the above example will be replaced by "showed evidence of no_congestive_heart_failure or no_pneumonia". Thus, our negation-aware processing step helps with the recognition of positive and negative mentions of a condition,

which we expect to enhance the predictive ability of our algorithm.

5.3.2 Transforming Text into Vectorized Features

For classification, we utilize two common standard text vector representations. A vectorized form of a hospital stay's aggregated notes was created using either the term frequency-inverse document frequency (TF-IDF) representation or a sequential word embedding representation.

Term Frequency–Inverse Document Frequency (TF-IDF representation)

The TF-IDF representation assigns weights to each unique word in the document. The weight accounts for the number of times a word appears in a document (or hospital-stay) and also adjusts for the frequency of the words in the overall corpus (across all hospital stays). Thus, words that occur more frequently in a document will have a higher impact, while rare words will have little influence. However, words that frequently appear across many documents will be less important. Under this representation, the sequence of words in the notes is not modeled as each document is viewed as a collection of words.

Sequential Word Embedding Representation

A major limitation of TF-IDF is the inability to preserve the word order in the text. To address this, the sequential word embedding representation assigns each word token a unique token number that preserves the word order in the text. A maximum number of words are specified for each document, and shorter documents are padded with zeros while longer ones are trimmed to the maximum number of words. These vectorized text representations were later passed to an embedding input layer where a dense representation of the word is learned.

5.3.3 PUI Classifiers

After curating the 2 versions of nursing notes (raw notes and negation-detected notes) and transforming the text to the appropriate vectorized representation, we predicted PUI incidence within each stay using only these features. We explored three different classifiers: logistic regression, random forest, and neural networks. Below, an overall view of each classifier is provided.

Logistic Regression: Interpretable and Intuitive

The logistic regression (LR) classifier is chosen since it is a straightforward and highly interpretable model. There is a one-to-one mapping from weights to features, which can be interpreted as the relative contribution (importance) of that feature to LR's decision toward the positive class. The TF-IDF feature representation is used as input for the LR classifier.

Random Forest: Trading Less Interpretability for Better Performance

The random forest (RF) classifier combines multiple learners for more accurate predictions. Due to its ensemble nature, it often achieves better predictive performance. RF also calculates feature importance based on the number and level of splits made with each feature across all the trees; however, the exact contribution is not readily apparent. The TF-IDF feature representation is also used as input for the RF classifier. In summary, applying RF on our PUI detection task can attain better performance at the expense of losing some interpretability.

Neural Networks: Many Parameters, Data Hungry

NN classifiers have been increasingly adopted for text data due to their stellar success [77]. We note that some NN architectures are more suitable than others for certain application domains and kinds of data. We fed the sequence of notes' words embedding vectors to a sequential model consisting of input word embedding, global max-pooling layer, and several dense layers, with the output layer yielding the probability of PUI incidence. We also tried the long short-term memory ($LSTM$)

network, given the sequential nature of the text. However, we excluded it from the results due to its poor performance.

5.4 Experimental Setup

Here we outline our experiments along with measures taken to ensure the robustness of our results, provide detailed parameters of classifiers, and describe our metric for evaluation. Finally, we report how feature importance pertaining to each classifier is leveraged to reveal keywords specific to its criteria for distinguishing PUI vs no PUI, where applicable. These extracted keywords for classifiers pave the way for more intuitive comparison across classifiers and more exploration of these keywords' compatibility with clinical characteristics of PUI.

5.4.1 Experiments Overview and Data Split

Our classification task is predicting the incidence of PUI in our test data given the training set. We used stratified sampling (i.e., the prevalence of PUI is maintained) to obtain three splits: 68% training, 12% validation, and 20% test data. To ensure our comparisons across the various predictive models and input data were generalizable, we repeated each experimental setting 30 times (i.e., over 30 different splits of training/validation/test). For a given split, all three classifiers were trained on the same training data, their hyperparameters were tuned on the same validation set, and made decisions about the labels of the same test data. Once the optimal hyperparameters for each classifier are found using the validation set, the classifier is retrained with these parameters on the training and validation data. For the final comparison, we report the average and variance of the predictive performance on the test data across all 30 splits.

Given our constructed versions of collapsed notes of each stay, untouched notes as

they're found in MIMIC-III, and processed notes using negation detection, and the three chosen classifiers Logistic Regression (LR), Random Forest (RF), and Neural Network (NN), we will have a total of 6 experimental settings.

5.4.2 Evaluation Metric

Our chosen cohort is unbalanced in terms of positive and negative cases of PUI. Thus, to truly capture the performance of our classifiers in different settings, we report the Area Under the Receiver Operating Characteristic Curve (AUC ROC) and F1 score on the test set.

5.4.3 Inferring Word Significance from Feature Importance

We assess the feature importance of specific words using the final trained model. Note that this is only applicable to LR and RF models, as NN requires additional methods to extract feature importance. For LR, we use the learned weights as indicators of feature importance. By sorting the weights in descending order, the features positively correlated with a PUI stay appear in the beginning, while the ones associated with the absence of PUI appear at the end. For RF, we report the feature importance that is calculated using the number and level of splits made on each feature across all decision trees.

5.5 Results and Discussion

We provide our evaluation results for the 2 versions of data (raw notes from MIMIC-III and the negation-aware notes) and the 3 chosen classifiers (LR, RF, and NN) and discuss the patterns observed. We first assess the merit of the negation detection step for PUI detection in multiple experimental settings and compare the performance of classifiers in predicting PUI. A comparison of extracted significant words with and

without applying negation detection is presented to illuminate the reasons for better performance of the negation detection step. Lastly, we discuss how relevant some of the significant words are to known medical contributors and/or certain comorbidities of PUI.

5.5.1 Impact of Negation Detection on AUC and F1 Score

Table 5.2 presents the average AUC and F1 score of LR, NN, and RF along with their standard deviation across the 30 splits. From the results, the negation detection step leads to AUC and F1 score improvements in all three classifiers. The greatest gain is for LR (around 3% and 5% for AUC and F1, respectively), followed by NN and RF. We did a one-sided paired t-test for each classifier to determine whether the improvement had a p-value below 0.05. The p-value for LR was less than machine precision for both metrics, while for NN and RF it was 0.2309 (0.0802) and 0.3826 (0.2599) (p-value for F1 is reported in parentheses). This further portrays the greatest utility of negation detection for boosting LR performance relative to improvement for NN and RF. Since RF performs random subsampling for each tree, these minority negative words are even less likely to be included in individual trees. However, in the NN and LR models, there is no random feature subsampling; therefore these same minority negative words have a higher chance of inclusion in the final model. This is also further supported by comparing the number of negative words among the important features when negation detection is used for these three classifiers.

5.5.2 Classifiers Performance Comparison

A comparison of the test AUC and F1 scores in Table 5.2 reveals that NN performs the worst. This is likely due to over-fitting as NN has an exponential number of parameters with respect to the number of layers and units. Given our small cohort, we anticipated NN to not perform well, as confirmed by our results.

Table 5.2: Average AUC and F1 score of classifiers with and without negation detection over 30 runs. * denotes a p-value < 0.05 under a one-sided paired t-test.

Classifier	**Average Test AUC *(SD)***		**Average Test F1 *(SD)***	
	Negation-Aware	w/o Negation Detection	Negation-Aware	w/o Negation Detection
Neural Networks (NN)	0.8462 *(0.0169)*	0.8440 *(0.0161)*	0.6252 *(0.0291)*	0.6189 *(0.0302)*
Logistic Regression (LR)*	0.9022 *(0.0120)*	0.8720 *(0.0155)*	0.6905 *(0.0188)*	0.6455 *(0.0248)*
Random Forest (RF)	0.9533 *(0.0086)*	0.9530 *(0.0071)*	0.7887 *(0.0226)*	0.7862 *(0.0219)*

Impact of Negation Detection on Extracted Features (Words) from Models

Next, we compare the extracted significant words' lists for the original notes and negation-aware notes. In particular, we look at the significance of words for the negative class in the two versions of data, and especially those containing the prefix *no_* in the negation-aware version. These are especially interesting since they highlight the efficacy of our proposed negation detection approach. The top 10 most influential words by feature importance alluding to the absence or presence of PUI are presented for both untouched notes and negation-aware notes. Table 5.3 summarizes the words for both LR and RF.

We first see that among the 10 most important words indicating no PUI in LR and RF, 5 and 4 words respectively are the direct product of the negation detection step, proving its utility. Furthermore, the words from the negation-aware version have comparatively higher weights than their untouched notes counterparts (-0.2566 vs. -0.1955 for LR, and 0.0067 vs. 0.00038 for RF). This means the 10 features have a higher correlation (e.g., for logistic regression, higher log-odds) with the outcome. Furthermore, some of the words in the top 10 words for the model trained on untouched notes are only non-specific general descriptors, which is rare in negation-aware versions (e.g. all words in RF untouched notes case except "ganx" and "fio2"

Table 5.3: Top 10 most important features (words) in different experimental settings.

Classifier	**Found only in**	**Indicating PUI**	**Words in the Set *(importance)***
LR	Negation-aware notes	Absence	{mso *(-0.3182)*, groundglass *(-0.2953)*, swanganz *(-0.2915)*, preoperative *(-0.2730)*, no_ectopy *(-0.2632)*, no_edema *(-0.2560)*, independent *(-0.2531)*, no_sob *(-0.2137)*, no_pneumothorax *(-0.2107)*, number *(-0.1918)*, no_pulmonary *(-0.1881)*}
LR	Untouched notes	Absence	{ganz *(-0.3563)*, mso4 *(-0.3431)*, lat *(-0.3431)*, ward *(-0.2286)*, lima *(-0.1443)*, hyperthermia *(-0.1169)*, pepcid *(-0.1156)*, neoplasm *(-0.1043)*, Sao2 *(-0.1023)*, pyrexia *(-0.1006)*}
RF	Negation-aware notes	Presence or Absence	{no_wound *(0.0016)*, apply *(0.0011)*, multipodus *(0.0011)*, swanganz *(0.0008)*, sch *(0.0005)*, clip *(0.0004)*, [no_skin, no_pneumothorax, unit, no_infection] *(all 0.0003)*}
RF	Untouched notes	Presence or Absence	{lat *(0.0007)*, ptitle *(0.0006)*, ganz *(0.0005)*, name *(0.0004)*, [followup, numeric, lastname, identifier] *(all 0.0003)*, [fi02, defined] *(all 0.0002)*}

are non-specific).

Inferred Salient Features and their Overlap with Leading Medical Factors of PUI

We investigate how closely the most significant contributing words resemble known medical covariates of PUI. After review by our diverse team of computer, nursing, biostatistics, and health informatics scientists, we present the high importance keywords that are aligned with the established evidence on PUI guidelines, including the definition, staging, Braden Scale, personalized algorithms, and root cause analysis template. Table 5.4 shows these keywords extracted from the model for different experimental settings of note version, classifier, and the direction of the words contribution. For example, the keyword *no_erythema* is consistent with the updated definition of "Stage 1 Pressure Injury: Non-blanchable erythema of intact skin" [25]. The keywords *swangaz* (abbreviation for "Swan-Ganz catheterization") and *no_tube*

indicate PUI related to medical devices [25]. Keywords such as *sedate* or *PACU* (post anesthesia care unit); *no_secretions* or *no_stool*; *independent*; *no_obstruction* and *multipodus* are related to the Braden risk categories of sensory perception, moisture, mobility or activity, nutrition as well as friction and shear respectively [12]. Also, Keywords *diuresis* and *multipodus* are consistent with recently identified predictive features for PUI in the intensive care unit [21]. Diabetes glycemic control indicated by *no_insulin*, *no_hypotension* and *no_infection* were also related to the risky comorbidities in the root cause analysis template [48] . Although some keywords with relatively high importance did not stand out in the past evidence, they could inform future PUI research directions. For example, *no_her* could indicate gender difference.

Table 5.4: Most medically meaningful keywords in different experimental settings.

Classifier	**Type of Words (or notes)**	**Indicating PUI**	**Most Medically Meaningful Keywords in the Set *(importance)***
LR	Only no_ .... words	Presence	{no_wound *(0.2951)*, no_erythema *(0.2303)*, no_skin *(0.1629)*, no_infection *(0.1420)*, no_obstruction *(0.1403)*, no_ct *(0.1291)*, no_secretions *(0.0850)*, no_lesions *(0.0728)*}
LR	Only no_ .... words	Absence	{no_edema *(-0.2560)*, no_stool *(-0.1241)*, no_pain *(-0.0923)*, no_diuresis *(-0.0744)*, no_hemorrhage *(-0.0677)*, no_bleed *(-0.0589)*, no_bleeding *(-0.0523)*}
LR	Only Negation-aware	Absence	{swanganz *(-0.2915)*, no_edema *(-0.2560)*, independent *(-0.2531)*, pacu *(-0.1765)*, bloodtinged *(-0.1314)*, no_stool *(-0.1241)*}
RF	Only no_ .... words	Presence or Absence	{no_wound *(0.0016)*, [no_skin, no_tube] *(0.0003)*, [no_infection, no_blood, no_insulin, no_erythema, no_hypotension] *(all 0.0002)*, [no_diuresis, no_abscess, no_pain, no_stool, no_edema, no_gtt] *(all 0.0001)*}
RF	Only Negation-aware	Presence or Absence	{no_wound *(0.0016)*, multipodus *(0.0010)*, [no_skin , no_infection , no_tube, no_insulin] *(all 0.0003)*, sedate *(0.0002)*}

5.6 Conclusion and Potential for HAPUI detection using Text

Our proposed method leverages collective notes acquired from the entire care team in one hospital stay for HAPUI detection for the first time. We proposed a negation detection step for notes that moves the presence or absence of medical conditions closer to their location in a sentence. Through experimental results with three representative classifiers for the PUI detection task, we showed the efficacy of the negation detection method in improving models' predictive performance. We further separated the keywords in notes and analyzed the keywords contributing the most to the detection of PUI and their encouraging overlap with medical knowledge on PUI. We also observed that many of the negated condition keywords were actually among the most important words for PUI detection and were compatible with nursing knowledge on PUI. Based on these promising results, we believe applying negation detection can also improve the performance of CrowdTeacher. Also, frequency of some of the words determined by the classifiers like edema and erythema can be specifically utilized for our future HAPUI detection tasks.

Chapter 6

Leveraging Unlabeled Samples for HAPUI Detection

Current HAPUI detection suffers from a major limitation – a high number of labeled annotations for reasonable performance. Obtaining high-quality annotations for the HAPUI detection task can be daunting. As a motivating example, for our experiments in Chapter 3, our nurse annotator spent more than 50 hours labeling 85 hospital admissions, which clearly shows that a high number of annotations is not practical for real-word adoption. Thus there is the need for leveraging semi-supervised techniques to reduce the annotation burden. Some of the known disadvantages of self-training are its sensitivity to the quality of the base classifier and its inability to correct already misclassified samples in the labeled set added in the previous iterations [75]. There are also special challenges that arise in self-training in the case of extremely unbalanced datasets [62]. For instance, a classifier trained using self-training may get exceedingly good at predicting majority class, while failing to distinguish minority class samples. Since in most applications, such as medicine and fraud detection [69], performance on the minority class is of higher importance, applying self-training without remedying these concerns is not a viable option for these applications. Given

these limitations, we propose to combine self-training with Co-teaching since Co-teaching can theoretically account for the uncertainty of the newly labeled samples in the training process and can also down weight the misclassified samples in the growing labeled set in each iteration to mitigate its negative impacts on performance. In this chapter, we propose a new self-training algorithm that reduces the annotation burden for HAPUI detection while addressing these challenges.

6.1 Our model

The main goal of our new model is to reduce the number of labels required for learning a robust classifier. The self-training algorithm introduced in Section 2.5 serves as the basis for our proposed algorithm. Given the downfalls of self-training, we propose to combine self-training with Co-teaching since Co-teaching can compensate for the uncertainty of the newly labeled samples in the training process and can also down weight the misclassified samples in the augmented labeled set in each iteration. However, applying self-training for unbalanced classification problems is challenging [62]. The minority class in the labeled set may be insufficiently small for the classifier to distinguish them from the other classes; therefore, the intermediate classifiers in self-training will have less predictive value for the minority class. This problem is exacerbated by the iterative nature of self-training, as the additionally labeled subset used to extend the training set in each iteration will contain relatively higher percentages of the majority class. This, in time, further decreases the prevalence of the minority class and, therefore, the classifier's performance with regard to the minority class. Alternatively, this can be viewed as the classifier overestimation of the certainty of its predictions on unlabelled samples from the majority class.

6.1.1 Self-training for Extremely Unbalanced Classes

Our preliminary experiments indicate that just using traditional approaches of dealing with unbalanced data, such as undersampling, alone, can not eradicate the overprediction of the majority class. We hypothesize this is due to the lower number of samples overall which prohibits the classifier from learning a generalizable enough feature representation for both classes, especially the minority class. To combat this, our method attempts to incorporate more instances of the minority class in the self-training process through three mechanisms including undersampling.

Undersampling of the majority class Recent research work suggest an oversampling of the minority class, or undersampling of the majority class is beneficial in self-training based algorithms [41]. To supply a more balanced dataset to the Co-teaching algorithm in each training, we randomly drop 10% of negative samples in each self-training round. We experimented with oversampling of the minority class, but found it ineffective when combating extreme imbalanced classes.

Monte Carlo dropout to differentiate certainty of the samples Monte Carlo dropout has been successfully used to boost the performance of neural networks [26]. Monte Carlo dropout provides a more accurate class probability prediction by applying multiple instances of dropout on the learned network to generate the estimated probability, compared to not having any dropout at the prediction time in the standard setting. We employ Monte Carlo drop out ten times to compute the prediction on both the unlabelled set and test set for our proposed algorithm. Because of the randomness involved in network units, the predicted probabilities across the majority class are more varied, making their ranking for the next iteration of self-training more meaningful. Our empirical result showed that this more meaningful ranking indeed mitigated the problem of classifier overestimation of its certainty for majority class prediction or failing to finely distinguish the confidence of the majority class samples during the self-training process.

Normalizing minority class certainties and prioritizing minority class samples over majority class samples with the same certainty During the initial iterations of self-training, the certainty of the minority class is significantly lower than that of the majority class. To ensure that at least most certain samples predicted by the current version of the classifier are added to the labeled section, we normalize both the certainty of the majority and minority class by their respective maximum certainty.

6.1.2 Algorithm Details

As mentioned earlier, the original self-training algorithm is unable to boost classifier performance in extremely unbalanced classes, when high-quality labels are not available, and when there is uncertainty in the labels [62, 75]. Thus we introduce the above three techniques to remedy these challenges and learn a more robust classifier.

Starting with our labeled set, $\mathbf{X}_L$ and unlabeled set $\mathbf{X}_U$, we first undersample a fraction, ω, of the majority-class samples in the labeled set to obtain samples used in classifier training, $(\mathbf{X}_{tr}, \mathbf{y}_{tr})$:

$$(\mathbf{X}_{tr}, \mathbf{y}_{tr}) \leftarrow Undersample((\mathbf{X}_L, \mathbf{y}_L), \omega) \tag{6.1}$$

We run the Co-teaching algorithm on this undersampled set to obtain the current model in the next step.

$$Model \leftarrow Co\_teaching((\mathbf{X}_{tr}, \mathbf{y}_{tr}), \epsilon, n_{mc}) \tag{6.2}$$

Next, our model generates pseudo-labels for the unlabeled set $\mathbf{X}_U$ using the trained model, i.e., we get class probabilities and classes for the unlabeled data while utilizing n_{mc} rounds of Monte Carlo dropout. Here ϵ denotes the dropout rate. We set the certainty of each sample as the maximum probability across the k classes. To increase

the chance of minority-class samples within the unlabeled set for addition to the labeled set, we normalize all classes' probabilities:

$$(\mathbf{P}_U, \mathbf{y}_U) \leftarrow Model(\mathbf{X}_U, \epsilon, n_{mc}) \tag{6.3}$$

$$c_i^{norm} = c_i / c_k^{max} \text{ where } c_k^{max} = \max(c_i) \text{ for } i \in U \text{ if } k = \arg\max_{k \in K} (P_{ik}) \tag{6.4}$$

We then pick the a_{top} most certain unlabeled samples while prioritizing the minority class in case of certainty ties to obtain the highly reliable samples ($(\mathbf{X}_{certain}, \mathbf{y}_{certain})$ and add them to the current labeled set while removing them from the unlabeled set of the current iteration.

$$(\mathbf{X}_{certain}, \mathbf{y}_{certain}) = (\mathbf{X}_U, \mathbf{y}_U)[\textbf{chosen\_idxs}] \text{ s.t } |chosen\_idxs| = a_{top} \tag{6.5}$$

$$(\mathbf{X}_L, \mathbf{y}_L) = Concatenate((\mathbf{X}_L, \mathbf{y}_L), (\mathbf{X}_{certain}, \mathbf{y}_{certain})) \tag{6.6}$$

$$(\mathbf{X}_U, \mathbf{y}_U) = (\mathbf{X}_U, \mathbf{y}_U).Remove((\mathbf{X}_{certain}, \mathbf{y}_{certain})) \tag{6.7}$$

This process of pseudo-labeling continues till the unlabeled set size reduces to less than a_{top}.

The model trained from the last iteration of the labeled set is used as the final model from our algorithm.

$$\hat{\mathbf{y}}_T \leftarrow Model_{final}(\mathbf{X}_T, \epsilon, n_{mc}) \tag{6.8}$$

Throughout the modified self-training algorithm, we always multiply gradients of different classes according to the relative count of classes. The details of the modified self-training can also be found in Algorithm 2.

Algorithm 2: Modified Self-training.

Input: Labeled Dataset $(\mathbf{X}_L, \mathbf{y}_L)$, Unlabeled samples $\mathbf{X}_U$, Test samples $\mathbf{X}_T$, # of Monte Carlo epochs n_{mc}, # of added unlabeled samples per epoch a_{top}, Dropout rate ϵ, Majority-class undersampling fraction ω, # of classes K
Output: Test predictions $\hat{\mathbf{y}}_T$
while $|\boldsymbol{X}_U| > a_{top}$ **do**
    Undersample majority-class samples:
    $(\mathbf{X}_{tr}, \mathbf{y}_{tr}) \leftarrow Undersample((\mathbf{X}_L, \mathbf{y}_L), \omega)$
    Run Co_teaching with Monte Carlo dropout on it:
    $Model \leftarrow Co\_teaching((\mathbf{X}_{tr}, \mathbf{y}_{tr}), \epsilon, n_{mc})$
    Get class probabilities and predicted classes for unlabeled samples:
    $(\mathbf{P}_U, \mathbf{y}_U) \leftarrow Model(\mathbf{X}_U, \epsilon, n_{mc})$
    Set certainty for unlabeled samples: $c_i \leftarrow \max_{k \in K}(P_{ik})\ \forall i \in U$
    Normalize certainties:
    $c_i^{norm} = c_i / c_k^{max}$ where $c_k^{max} = \max(c_i)$ for $i \in U$ if $k = \arg\max_{k \in K}(P_{ik})$
    Pick the a_{top} most certain points prioritizing minority class for ties:
    $(\mathbf{X}_{certain}, \mathbf{y}_{certain}) = (\mathbf{X}_U, \mathbf{y}_U)[\textbf{chosen\_idxs}]$ s.t $|chosen\_idxs| = a_{top}$
    Add these highly reliable samples to the labeled dataset:
    $(\mathbf{X}_L, \mathbf{y}_L) = Concatenate((\mathbf{X}_L, \mathbf{y}_L), (\mathbf{X}_{certain}, \mathbf{y}_{certain}))$
    Shrink unlabeled dataset:
    $(\mathbf{X}_U, \mathbf{y}_U) = (\mathbf{X}_U, \mathbf{y}_U).Remove((\mathbf{X}_{certain}, \mathbf{y}_{certain}))$
end
Run Co_teaching with Monte Carlo dropout on final $\mathbf{X}_L$ and get $\hat{\mathbf{y}}_T$:
$(\mathbf{X}_{tr}, \mathbf{y}_{tr}) \leftarrow Undersample((\mathbf{X}_L, \mathbf{y}_L), \omega)$
$Model_{final} \leftarrow Co\_teaching((\mathbf{X}_{tr}, \mathbf{y}_{tr}), \epsilon, n_{mc})$ $\hat{\mathbf{y}}_T \leftarrow Model_{final}(\mathbf{X}_T, \epsilon, n_{mc})$
Return $\hat{\mathbf{y}}_T$

6.2 Experimental Settings

We test our model on two real-world prediction tasks, HAPUI and Length-of-Stay in MIMIC-III.

6.2.1 HAPUI

We define the following cohort and starting with 8237 labels for 5742 samples, we examine how much we can lower the number of labels without significantly compromising performance.

1. Admissions whose patients are younger than 65 years old.

2. Admissions that were admitted through the emergency department.

3. Admission with a duration greater than ten days.

This yielded 5742 samples total, with 822 admissions marked for the presence of HAPUI. To ensure the most confident labels are being assigned for annotation generation, we used the admissions with the same labels based on both our golden criteria and CANTRIP criteria introduced in Chapter 3.

For generating different experimental settings, for the given training dataset from a 80/20 split of our cohort, we varied the ratio of samples used as unlabeled data in the interval [0.03, 0.166] in 0.03 increments. We annotated the labeled samples based on the following distribution across the number of labels using 5 annotators $[0.51(1-\tau), 0.25(1-\tau), 0.13(1-\tau), 0.07(1-\tau), 0.04(1-\tau)]$. Here τ denotes the fraction of unlabeled samples. The average number of annotations per labeled sample is 1.87 and the same across all settings. Notice as unlabeled sample ratio goes up, the labeled set is shrunk too so the total number of annotations is reduced too.

6.2.2 Length-of-stay

For the cohort, we used the same one described in Section 4.3.3. To create annotations, we varied the ratio of samples used as unlabeled data in the interval [0.03, 0.6] in 0.03 increments. We set the labels for the labeled samples based on the following distribution across different number of labels using 5 annotators $[0.15(1-\tau), 0.2(1-\tau), 0.35(1-\tau), 0.25(1-\tau), 0.05(1-\tau)]$. Here τ shows the fraction of unlabeled samples. The average number of annotations per labeled sample is 2.87 and is the same across all settings. Note tha as unlabeled sample ratio goes up, the labeled set is shrunk too so the total number of annotations is reduced too.

6.3 Results for HAPUI Detection

We compare the performance of our model, with the base classifier comprised of a single layer neural network of 20 units shared across all methods, Vanilla self-training with the base classifier and Co-teaching with no self-training.

Figure 6.1 demonstrates that our model, the modified self-training plus Co-teaching, is performing better than the above baselines across different ratios of unlabeled set and across varying total numbers of labels in terms of AUPRC. As the number of labeled set decreases on the horizontal axis, the base classifier AUPRC plummets. However, despite this decreasing AUPRC, we still observe that our model continues to boost performance, compared to using Vanilla self-training on the base classifier or only using the Co-teaching algorithm without leveraging unlabeled data.

Another pattern is that initially with more labeled samples, self-training does better than Co-teaching, but as this number goes down and the classifier on the labeled set becomes weaker, from 0.1 onwards, Co-teaching is the better method for prediction. We hypothesize this is due to the lower accuracy of the pseudo labels as the performance of classifiers on labeled set becomes worse, which makes self-training on inaccurate pseudo-labels harmful rather than useful. Notice how this is not the case on our combined modified self-training and Co-teaching since Co-teaching helps with reducing the impact of inaccurate labels within the pseudo labels set.

6.4 Results for LOS Prediction

We also evaluated our model across different ratios of samples used in the unlabeled set for the length of stay prediction task introduced in Section 4.3.3.

In Figure 6.2, we change the ratio of samples used in the unlabeled set and self-training from 3% to 60% in 3% increments. The vertical axis shows the test AUPRC. We can see that regardless of the ratio of the unlabeled set, our modified self-training

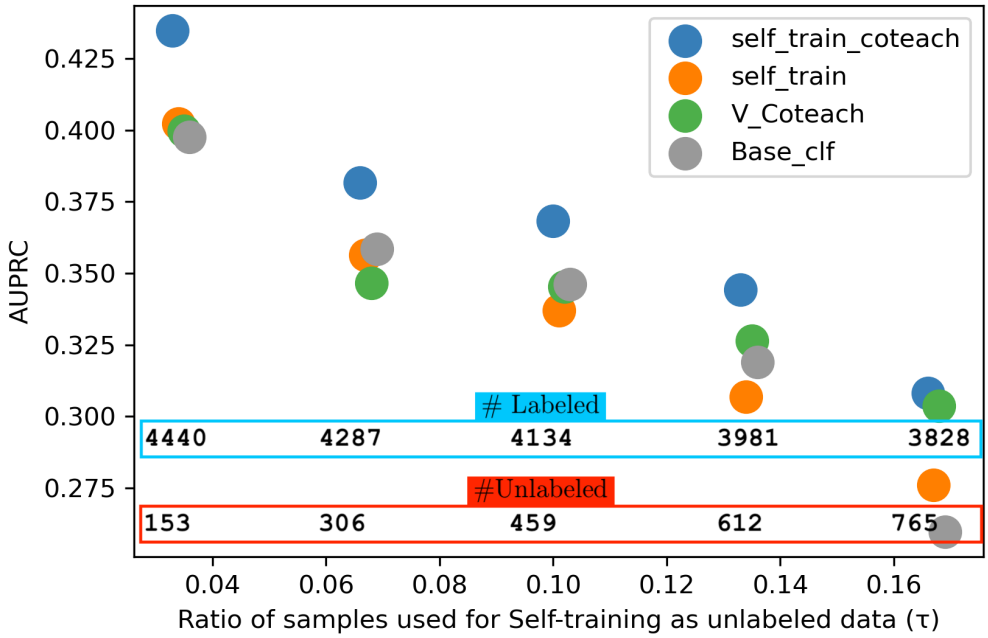


Figure 6.1: Modified Self-training + Co-Teaching for HAPUI detection.

combined with Co-teaching performs better than base classifier with only the labeled set, Co-teaching with only the labeled data, and Vanilla self-training with base classifier using both unlabeled and labeled data. Furthermore, we can observe by having only annotations for half of the data, we can still get a reasonable AUPRC of 0.6.

Unlike the HAPUI task, here self-training is performing worse than Co-teaching and base classifier regardless of the ratio of unlabeled samples. We believe this is due to the higher AUPRC compared to the HAPUI task, which makes enhancements more challenging, and therefore to achieve performance boost, the floor threshold for accuracy of pseudo labels is higher, which makes the Vanilla self-training the weakest method at almost all points.

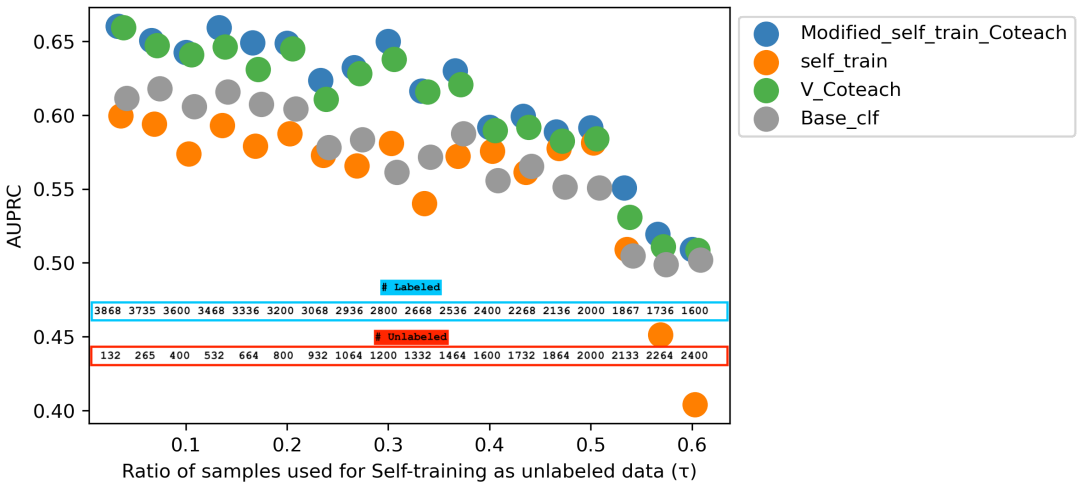


Figure 6.2: Modified self-training + Co-Teaching for LOS Prediction.

6.5 Conclusion

We first introduced the challenges arising in self-training with extreme unbalanced classes settings and uncertain classes for labeled sets. We introduced the combination of three countermeasures that enable self-training to boost performance in these situations, Monte Carlo dropout, and certainly normalization and prioritization for minority class combined with undersampling. Finally, we formally introduced our new algorithm and showed its utility in increasing AUPRC on both HAPUI detection and the length of stay prediction tasks. Our work is the first to employ self-training for extreme imbalanced datasets using uncertain annotations and proposes the mixture of several techniques to accommodate unbalanced classes and uncertain labels using the modified self-training plus Co-teaching algorithm.

Chapter 7

Conclusion and Future work

In this thesis, we explored how EHRs can be efficiently used for detecting HAPUI. We first presented the challenges surrounding establishing ground truth from multiple EHR sources and their conflicts. We proposed a novel standardized cohort definition that is more faithful to the clinical guidelines compared to the already available alternatives, which makes the comparison of algorithms detecting HAPUI using EHRs possible. In the next chapter, we introduced CrowdTeacher a novel uncertainty-aware sample-specific perturbation scheme that performs better than the existing baselines for learning with uncertain crowdsourcing labels, a practical formulation for HAPUI detection. To glean better feature representations, we analyzed how hospital notes can be leveraged for HAPUI detection and whether the words classifiers most relied on, overlapped with known literature on HAPUI. In the last chapter, we devised a modified Self-training algorithm in conjunction with Co-teaching to show how unlabeled samples can be beneficial for HAPUI detection by decreasing the annotation costs.

There are several future directions based on this work. For Chapter 5, we note that there exists a tool extension, ConText, that provides more accurate identification of medical terms and also provides uncertainty associated with the negation

detection. Thus instead of hard classification of negated phrases, we can incorporate the estimated probability of negation of the conditions into the learning process to further enhance the prediction. In addition, one can also consider further extending our modified self-training algorithm to leverage crowdsourcing uncertainty. Finally, another potential area of future work is to explore the use of even fewer number of labels to learn a robust classifier for HAPUI.

Bibliography

[1] 5. how do we measure our pressure ulcer rates and practices? https://www.ahrq.gov/patient-safety/settings/hospital/resource/pressureulcer/tool/put5.html. Accessed: 2021-4-5.

[2] CMS guideline for LTCH quality reporting. https://www.cms.gov/Medicare/Quality-Initiatives-Patient-Assessment-Instruments/LTCH-Quality-Reporting/Downloads/LTCH-QRP-Manual-Version-40-Effective-July-1-2zip, . Accessed: 2021-4-7.

[3] Long-Term care hospital quality reporting program measure calculations and reporting user's manual version 3.1. https://www.cms.gov/Medicare/Quality-Initiatives-Patient-Assessment-Instruments/LTCH-Quality-Reporting/Downloads/LTCH-Measure-Calculations-and-Reporting-Userspdf, . Accessed: 2021-5-1.

[4] ICD-9-CM diagnosis and procedure codes: Abbreviated and full code titles. https://www.cms.gov/Medicare/Coding/ICD9ProviderDiagnosticCodes/codes. Accessed: 2021-4-30.

[5] NPIAP PUI awareness fact sheet. https://cdn.ymaws.com/npiap.com/resource/resmgr/npiap_pru_awareness_fact_she.pdf, . Accessed: 2021-4-20.

[6] NPIAP pressure injury staging description. https://npiap.com/page/PressureInjuryStages, . Accessed: 2021-3-27.

[7] 2019 guideline QRG E-Version (NPIAP). https://npiap.com/page/Guidelines. Accessed: 2021-5-1.

[8] Agency for Healthcare Research and Quality. Preventing Pressure Ulcers in Hospitals. Content last reviewed October 2014. URL https://www.ahrq.gov/patient-safety/settings/hospital/resource/pressureulcer/tool/index.html.

[9] Shadi Albarqouni, Christoph Baur, Felix Achilles, Vasileios Belagiannis, Stefanie Demirci, and Nassir Navab. Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE transactions on medical imaging*, 35(5):1313–1321, 2016.

[10] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27:3365–3373, 2014.

[11] Ofir Ben-Assuli, Doron Sagi, Moshe Leshno, Avinoah Ironi, and Amitai Ziv. Improving diagnostic accuracy using EHR in emergency departments: A simulation-based study. *J. Biomed. Inform.*, 55:31–40, June 2015.

[12] N Bergstrom, B J Braden, A Laguzza, and V Holman. The Braden Scale for Predicting Pressure Sore Risk. *Nursing research*, 36(4):205–210, 1987. ISSN 0029-6562 (Print).

[13] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019.

[14] Joyce M Black, Janet E Cuddigan, Maralyn A Walko, L Alan Didier, Maria J Lander, and Maureen R Kelpe. Medical device related pressure ulcers in hospitalized patients. *International wound journal*, 7(5):358–365, oct 2010. ISSN 1742-481X (Electronic). doi: 10.1111/j.1742-481X.2010.00699.x.

[15] Harold Brem, Jason Maggi, David Nierman, Linda Rolnitzky, David Bell, Robert Rennert, Michael Golinko, Alan Yan, Courtney Lyder, and Bruce Vladeck. High cost of stage IV pressure ulcers. *American journal of surgery*, 200(4):473–477, oct 2010. ISSN 1879-1883 (Electronic). doi: 10.1016/j.amjsurg.2009.12.021.

[16] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[17] Centers for Medicare and Medicaid Services. Readmissions Reduction Program., 2015.

[18] Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310, 2001.

[19] Insook Cho, Ihnsook Park, Eunman Kim, Eunjoon Lee, and David W Bates. Using EHR data to predict hospital-acquired pressure ulcers: a prospective study of a Bayesian Network model. *International journal of medical informatics*, 82 (11):1059–1067, nov 2013. ISSN 1872-8243 (Electronic). doi: 10.1016/j.ijmedinf.2013.06.012.

[20] Jill Cox, Marilyn Schallom, and Christy Jung. Identifying risk factors for pressure injury in adult critical care patients. *Am. J. Crit. Care*, 29(3):204–213, May 2020.

[21] Eric Cramer, Martin Seneviratne, Husham Sharifi, Alp Ozturk, and Tina Hernandez-Boussard. Predicting the Incidence of Pressure Ulcers in the Intensive Care Unit Using Machine Learning. *eGEMs (Generating Evidence & Methods to improve patient outcomes)*, 7, sep 2019. doi: 10.5334/egems.307.

[22] Eric M Cramer, Martin G Seneviratne, Husham Sharifi, Alp Ozturk, and Tina Hernandez-Boussard. Predicting the incidence of pressure ulcers in the intensive care unit using machine learning. *EGEMS (Wash DC)*, 7(1):49, September 2019.

[23] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[24] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pages 469–478. ACM, 2012.

[25] Laura E Edsberg, Joyce M Black, Margaret Goldberg, Laurie McNichol, Lynn Moore, and Mary Sieggreen. Revised National Pressure Ulcer Advisory Panel Pressure Injury Staging System: Revised Pressure Injury Staging System. *Journal of wound, ostomy, and continence nursing : official publication of The Wound, Ostomy and Continence Nurses Society*, 43(6):585–597, 2016. ISSN 1528-3976 (Electronic). doi: 10.1097/WON.0000000000000281.

[26] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016.

[27] Alex Gaunt, Diana Borsa, and Yoram Bachrach. Training deep neural nets to aggregate crowdsourced responses. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence. AUAI Press*, page 242251, 2016.

[28] Travis R Goodwin and Dina Demner-Fushman. A customizable deep learning model for nosocomial risk prediction from critical care notes with indirect supervision. *J. Am. Med. Inform. Assoc.*, 27(4):567–576, April 2020.

[29] Melody Y Guan, Varun Gulshan, Andrew M Dai, and Geoffrey E Hinton. Who said what: Modeling individual labelers improves classification. *arXiv preprint arXiv:1703.08774*, 2017.

[30] Fábio Gusmão Vicente, Frederico Polito Lomar, Christian Mélot, and Jean-Louis Vincent. Can the experienced ICU physician predict ICU length of stay and outcome better than less experienced colleagues? *Intensive Care Med.*, 30(4): 655–659, April 2004.

[31] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.

[32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[33] Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 2492–2500, 2014.

[34] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and

Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[35] David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, pages 1953–1961, 2011.

[36] Hyun-Chul Kim and Zoubin Ghahramani. Bayesian classifier combination. In *Artificial Intelligence and Statistics*, pages 619–627, 2012.

[37] Junnan Li, Richard Socher, and Steven C H Hoi. DivideMix: Learning with noisy labels as semi-supervised learning. February 2020.

[38] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment*, 8(4):425–436, 2014.

[39] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1187–1198. ACM, 2014.

[40] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. A survey on truth discovery. *ACM Sigkdd Explorations Newsletter*, 17(2): 1–16, 2016.

[41] Yunru Liu, Tingran Gao, and Haizhao Yang. SelectNet: Learning to sample from the wild for imbalanced data training. In Jianfeng Lu and Rachel Ward, editors, *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pages 193–206, Princeton University, Princeton, NJ, USA, 20–24 Jul 2020. PMLR. URL `http://proceedings.mlr.press/v107/liu20a.html`.

[42] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. May 2017.

[43] Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pages 410–418, 2013.

[44] Elizabeth McInnes, Asmara Jammali-Blasi, Sally E M Bell-Syer, Jo C Dumville, Victoria Middleton, and Nicky Cullum. Support surfaces for pressure ulcer prevention. *Cochrane Database Syst. Rev.*, (9):CD001735, September 2015.

[45] Pooya Mobadersany, Safoora Yousefi, Mohamed Amgad, David A. Gutman, Jill S. Barnholtz-Sloan, José E. Velázquez Vega, Daniel J. Brat, and Lee A. D. Cooper. Predicting cancer outcomes from histology and genomics using convolutional networks. *Proceedings of the National Academy of Sciences*, 115(13): E2970–E2979, 2018. ISSN 0027-8424. doi: 10.1073/pnas.1717139115. URL `https://www.pnas.org/content/115/13/E2970`.

[46] Z E H Moore and D Patton. Risk assessment tools for the prevention of pressure ulcers. *Cochrane Database of Systematic Reviews*, (1), 2019. ISSN 1465-1858. doi: 10.1002/14651858.CD006471.pub4. URL `https://doi.org//10.1002/14651858.CD006471.pub4`.

[47] Antonio Paulo Nassar, Jr and Pedro Caruso. ICU physicians are unable to accurately predict length of stay at admission: a prospective study. *Int. J. Qual. Health Care*, 28(1):99–103, February 2016.

[48] National Pressure Ulcer Advisory Panel. NPUAP Pressure Ulcer Root Cause Analysis Template. URL `http://www.hret-hiin.org/resources/display/npuap-pressure-ulcer-root-cause-analysis-template`.

[49] National Pressure Ulcer Advisory Panel; European Pressure Ulcer Advisory Panel. Prevention and Treatment of Pressure Ulcers: Clinical Practice Guideline, 2012.

[50] National Pressure Ulcer Advisory Panel (U.S.). *Prevention and Treatment of Pressure Ulcers: Clinical Practice Guideline*. September 2014.

[51] National Quality Forum. National Voluntary Consensus Standards for Developing a Framework for Measuring Quality for Prevention and Management of Pressure Ulcers. Technical report, 2011.

[52] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*, 2019.

[53] Viet-An Nguyen, Peibei Shi, Jagdish Ramakrishnan, Udi Weinsberg, Henry C. Lin, Steve Metz, Neil Chandra, Jane Jing, and Dimitris Kalimeris. *CLARA: Confidence of Labels and Raters*, page 2542–2552. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450379984. URL `https://doi.org/10.1145/3394486.3403304`.

[54] N. Patki, R. Wedge, and K. Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, Oct 2016. doi: 10.1109/DSAA.2016.49.

[55] Nitin Namdeo Pise and Parag Kulkarni. A survey of semi-supervised learning methods. In *2008 International Conference on Computational Intelligence and Security*, volume 2, pages 30–34, 2008. doi: 10.1109/CIS.2008.204.

[56] Otavio T Ranzani, Evelyn Senna Simpson, André M Japiassú, and Danilo Teixeira Noritomi. The Challenge of Predicting Pressure Ulcers in Critically Ill

Patients. A Multicenter Cohort Study. *Annals of the American Thoracic Society*, 13(10):1775–1783, 2016. doi: 10.1513/AnnalsATS.201603-154OC. URL https://doi.org/10.1513/AnnalsATS.201603-154OC.

[57] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

[58] Madhuri Reddy, Sudeep S Gill, and Paula A Rochon. Preventing pressure ulcers: a systematic review. *JAMA*, 296(8):974–984, August 2006. ISSN 1538-3598 (Electronic). doi: 10.1001/jama.296.8.974.

[59] Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. 32(1), 2018.

[60] CA Russo, C Steiner, and W Spector. Hospitalizations Related to Pressure Ulcers Among Adults 18 Years and Older, 2006: Statistical Brief #64. 2008 Dec. In: Healthcare Cost and Utilization Project (HCUP) Statistical Briefs. Rockville (MD): Agency for Healthcare Research and Quality. URL https://www.ncbi.nlm.nih.gov/books/NBK54557/.

[61] Nihal Soans, Ehsan Asali, Yi Hong, and Prashant Doshi. Sa-net: Robust state-action recognition for learning from observations. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2153–2159. IEEE, 2020.

[62] Ana Stanescu and Doina Caragea. An empirical study of ensemble-based semi-supervised learning approaches for imbalanced splice site datasets. *BMC Syst. Biol.*, 9 Suppl 5:S1, September 2015.

[63] Jürgen Stausberg, Nils Lehmann, Knut Kröger, Irene Maier, Wolfgang Niebel, and interdisciplinary decubitus project. Reliability and validity of pressure ulcer diagnosis and grading: an image-based survey. *Int. J. Nurs. Stud.*, 44(8):1316–1323, November 2007.

[64] F. Tahmasebian, L. Xiong, M. Sotoodeh, and V. Sunderam. Edgeinfer: Robust truth inference under data poisoning attack. In *2020 IEEE International Conference on Smart Data Services (SMDS)*, pages 45–52, 2020. doi: 10.1109/SMDS49396.2020.00013.

[65] Farnaz Tahmasebian, Li Xiong, Mani Sotoodeh, and Vaidy Sunderam. Crowdsourcing under data poisoning attacks: A comparative study. In Anoop Singhal and Jaideep Vaidya, editors, *Data and Applications Security and Privacy XXXIV*, pages 310–332, Cham, 2020. Springer International Publishing. ISBN 978-3-030-49669-2.

[66] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. February 2019.

[67] Catherine VanGilder, Stephanie Amlung, Patrick Harrison, and Stephanie Meyer. Results of the 2008-2009 International Pressure Ulcer Prevalence Survey and a 3-year, acute care, unit-specific analysis. *Ostomy/wound management*, 55 (11):39–45, nov 2009. ISSN 1943-2720 (Electronic).

[68] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164. ACM, 2014.

[69] Jue Wang and Anoop Cherian. Gods: Generalized one-class discriminative subspaces for anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8201–8211, 2019.

[70] Shirly Wang, Matthew B. A. McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C. Hughes, and Tristan Naumann. Mimic-extract. *Proceedings*

of the ACM Conference on Health, Inference, and Learning, Apr 2020. doi: 10.1145/3368555.3384469. URL http://dx.doi.org/10.1145/3368555.3384469.

[71] Shirley Moore Waugh and Sandra Bergquist-Beringer. Inter-rater agreement of pressure ulcer risk and prevention measures in the national database of nursing quality indicators (ndnqi). *Research in nursing & health*, 39(3):164–174, 2016.

[72] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[73] Kathy T Whittington and Robin Briones. National Prevalence and Incidence Study: 6-year sequential acute care data. *Advances in skin & wound care*, 17(9):490–494, 2004. ISSN 1527-7941 (Print). doi: 10.1097/00129334-200411000-00016.

[74] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*, pages 7335–7345, 2019.

[75] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* -, Morristown, NJ, USA, 1995. Association for Computational Linguistics.

[76] Li'ang Yin, Jianhua Han, Weinan Zhang, and Yong Yu. Aggregating crowd wisdoms with label-aware autoencoders. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1325–1331. AAAI Press, 2017.

[77] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent

trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[78] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *CoRR*, abs/1710.09412, 2017. URL `http://arxiv.org/abs/1710.09412`.

[79] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9294–9303, 2020.

[80] Dengyong Zhou, Sumit Basu, Yi Mao, and John C Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in neural information processing systems*, pages 2195–2203, 2012.

[81] Yao Zhou and Jingrui He. Crowdsourcing via tensor augmentation and completion. In *IJCAI*, pages 2435–2441, 2016.

[82] Yao Zhou, Lei Ying, and Jingrui He. Multic2: an optimization framework for learning from task and worker dual heterogeneity. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 579–587. SIAM, 2017.