Automating Weak Label Generation for Data Programming with Clinicians in the Loop

Jean Park*, Sydney Pugh*, Kaustubh Sridhar, Mengyu Liu, Navish Yarna,
Ramneet Kaur, Souradeep Dutta, Elena Bernardis, Oleg Sokolsky, and Insup Lee
University of Pennsylvania
Philadelphia, PA, USA
{hlpark, sfpugh, duttaso, elber, lee}@seas.upenn.edu

***Abstract*—Deep learning with its unique ability to mine patterns in complex data has the ability to transform smart health. However, large Deep Neural Networks (DNNs) are often data hungry and need high-quality labeled data in copious amounts for learning to converge. This is a challenge in the field of medicine since high quality labeled data is often scarce. Data programming has been the ray of hope in this regard, since it allows us to label unlabeled data using multiple weak labeling functions. Such functions are often supplied by a domain expert. Data-programming can combine multiple weak labeling functions and suggest labels better than simple majority voting over the different functions. However, it is not straightforward to express such weak labeling functions, especially in high-dimensional settings such as images and time-series data. What we propose in this paper is a way to bypass this issue, using distance functions. In high dimensional spaces, it is easier to find meaningful distance metrics which can generalize across different labeling tasks. We propose an algorithm that queries an expert for labels of a few representative samples of the dataset. These samples are carefully chosen by the algorithm to capture the distribution of the dataset. The labels assigned by the expert on the representative subset induce a labeling on the full dataset, thereby generating weak labels to be used in the data programming pipeline. In our medical time series case study, labeling a subset of 50 to 130 out of 3,265 samples showed 17-28% improvement in accuracy and 13-28% improvement in F1 over the baseline using clinician-defined labeling functions. In our medical image case study, labeling a subset of about 50 to 120 images from 6,293 unlabeled medical images using our approach showed significant improvement over the baseline method, Snuba, with an increase of approximately 5-15% in accuracy and 12-19% in F1 score.**

***Index Terms*—data programming, machine learning, weak supervision.**

I. Introduction

Deep learning has shown significant promise in various applications within the field of medicine, ranging from diagnostic imaging to drug discovery [1]–[3]. However, its widespread adoption and effectiveness is constrained by several challenges, especially the scarcity of labeled data [4], [5]. Medical data, especially for rare conditions, is often limited [6]. Labeling the data requires expert-level knowledge, which can be time-consuming and expensive. Moreover, the rarity of certain medical conditions means that there are fewer instances or cases available for study, making it difficult to gather a sufficiently large dataset that deep learning algorithms require for effective training and validation. This scarcity of data can lead to sub-optimal models that do not adequately represent the full spectrum of the condition, potentially leading to less effective or even misleading outcomes when these models are applied in a clinical setting. Unfortunately, the assumption about the presence of abundant high quality labeled samples in medical settings, is often not the main concern when designing increasingly complex deep learning architectures. Thus it is imperative that the research community pays more attention to address this challenge.

To remedy the data scarcity challenge, data programming has emerged as a potential alternative [7]–[9]. Data programming is a paradigm that aims to simplify and scale up the process of creating labeled training data, which is essential for supervised learning [7]. The dominant learning paradigm in medical AI, manual labeling of data, is often prohibitively expensive and time-consuming, especially for large datasets. Data programming offers a more cost-effective and scalable alternative [10]. It allows domain experts to encode their knowledge in the form of *rule of thumb* functions also known as weak labeling functions. Data programming uses these weak supervision signals to generate labels, which can be used for training a machine learning model, effectively bridging the gap between domain expertise and machine learning.

In reality, such weak labeling functions are heuristic or algorithmic rules that can assign labels to data points [7], [11]. These functions are qualified as "weak" as a reminder to the fact that they are not as accurate or consistent as an expert labeling function can be. However, such heuristic rules are often present as domain knowledge and are amenable to being encoded easily. Developing weak labeling functions for data programming in machine learning involves a combination of domain expertise, data exploration, and heuristic approaches. Typically, experts provide valuable insights and identify specific patterns or indicators relevant to the labeling task. Alongside expert consultation, exploratory data analysis plays a vital role. It helps in uncovering hidden patterns, correlations, or frequent occurrences within the data, which can inform the creation of effective labeling functions.

However, it is challenging to construct weak labeling functions for high-dimensional data such as images and long horizon time series data. In order to be useful, the weak labeling function almost has to act like a decent classifier.

*Both authors contributed equally to this work.

It is well known that high-dimensional data samples contain vast amounts of information and variability. Therefore, it is hard to manually identify patterns and features across the dimensions. For instance, complicated features like texture may comprise subtle patterns that are not immediately obvious to the programmer. Even though experts can identify subtle textures, articulating this knowledge in a formal way that can be used to train machine learning models is challenging. The tacit knowledge that experts possess is not always easily transferable to weak-labeling functions.

To address this above challenge, we propose distance-based weak labels generation for high-dimensional data. Specifically, the contribution of this work are as follows:

1) We show that using carefully chosen labeled samples to construct weak labels can significantly improve labeling accuracy. We demonstrate this with a thorough comparison against the weak label generation tool, *Snuba*, for an equivalent number of labeled samples.
2) We propose a novel algorithm which involves clinicians in the loop to label prototypical samples. We believe that our novel approach paves the way to clinician-AI collaboration in medical settings.
3) We evaluate our algorithm on two challenging high-dimensional datasets – (1) a time-series dataset consisting of 3,265 low SpO_2 alarms collected as part of a study at a major hospital and (2) a dataset containing 6,293 de-identified medical images uploaded to a major U.S. hospital's electronic medical records labeled for specific body parts. The experimental results show the generated labeling function outperform the baselines by sufficient margins on both datasets.

II. Related Work

A. Clinical Trials for Labeled Medical Datasets

The gold standard for constructing large labeled medical datasets is a clinical trial, in which data is collected and manually labeled by clinical experts. Unfortunately, clinical trials require substantial time and effort, involving obtaining regulatory approvals, recruiting a diverse patient cohort, and labor-intensive and expensive manual labeling. For example, to create a dataset for training physiologic alarm suppression algorithms, nurses may need to review video feeds of patients to determine the actionability of alarms [12]. Such manual analysis can be extensive given the significantly high volume of physiologic monitoring alarms in hospitals, many of which are not actionable [13]. Our work is not meant to replace clinical trials; instead, we aim to reduce the amount of time spent manually labeling trial data by providing an semi-automated approach to labeling the data with the clinician in the loop.

B. Data Programming for Labeled Medical Datasets

Recently, a novel method for efficiently and affordably annotating data has been introduced, termed data programming [7]. This approach is especially relevant in the field of healthcare and is based on utilizing basic, quantitative insights to link data with its corresponding labels. For example, a healthcare professional might note that an alert should be highly prioritized if it concerns a patient over 65 years old with a heart rate exceeding 120 beats per minute for more than a minute. These observations are converted into algorithms known as *labeling functions*, which may be imperfect, sporadically inaccurate, or even inconsistent. These functions either assign a specific class label to data or opt to 'abstain' from labeling. By applying a diverse array of these labeling functions to an unlabeled dataset, data programming models generate labels for each data entry as probability distributions across the possible labels. The weak label for a data entry is then determined as the label with the highest probability, with confidence matching this probability.

The concept of data programming has gained significant attention as a solution for efficiently managing and labeling large datasets. An established technique in this area is Snorkel [11], [14], [15], which calculates the most effective weight for each labeling function through a generative graphical model, incorporating prior knowledge about the distribution of classes. This incorporation is crucial as it allows Snorkel to more accurately weigh the importance and relevance of each labeling function based on the known class distributions. By doing so, Snorkel effectively calibrates the labeling functions to better reflect the real-world scenarios they are intended to model, enhancing the accuracy and reliability of the labels generated for the dataset. This feature is particularly valuable in scenarios where class distributions are uneven or where some classes are significantly more prevalent than others, a common occurrence in many real-world datasets.

Labeling functions in weakly-supervised data labeling systems, such as Snorkel, are typically characterized by their inherent noise and variable error rates [7], [16]. This means that these functions, which are designed to assign labels to data points based on specific rules or heuristics, often produce labels that are not always accurate. The error rates can vary significantly between different labeling functions, depending on the complexity of the rules they apply and the quality of the underlying data they process. Moreover, these labeling functions may sometimes generate conflicting labels for the same data points [17]. This conflict arises because each labeling function operates independently, based on its own set of criteria or heuristics, without considering the outputs of other functions. For instance, one labeling function might label a data point as belonging to one class based on a particular feature or pattern, while another function might assign a different class to the same data point based on a different feature or pattern.

To address this issue, researchers explored the possibility of developing labeling generation methods that aggregate the noisy votes from multiple labeling functions [18]–[20]. Researchers in [7], [11], [20] propose a two-stage approach for weak labeling which using the training labels to train an end model for downstream tasks. Some researchers have attempted to develop a one-stage approach which integrates the label model with the end model [18], [19]. It seeks to unify

the process of labeling and model training, thereby reducing the complexity and potential errors associated with multi-stage processes.

Developing labeling functions for data programming can be challenging for domain experts, particularly for data such as time series and images. As a result, recent work have explored automated methods for generating labeling functions [21]–[24]. In the literature, the closest effort similar to our proposed approach is *Snuba*. It uses a small set of labeled examples (i.e., a validation dataset) to automatically generate and refine such heuristic functions [21]. These functions are then applied to unlabeled data to create a training set. Snuba iteratively improves the functions based on their performance, thereby enhancing the quality of the labeled data over time. Snuba can help identifying the most informative features and patterns in the data and help creating new heuristic weak labeling functions which is feasible when there is limited access to domain expertise for writing labeling functions.

III. Problem Formulation

Assume that the inputs to be labeled belong to the space $\mathcal{X}$ and the possible labels are in the finite set $\mathcal{Y}$. We are given an unlabeled dataset $\mathcal{D}_u := \{x_i\}_{i=1}^{N_u}$, such that each sample $x_i \in \mathcal{X}$ has a unique but unknown label in $\mathcal{Y}$. A vast majority of real world health data falls in this category. Note that in reality there exists an unknown oracle classifier $\mathcal{C} : \mathcal{X} \mapsto \mathcal{Y}$, which is capable of generating the ground truth labels $y_i = \mathcal{C}(x_i)$. However, $\mathcal{C}$ and therefore y_i is unknown.

Next, we discuss the concept of *weak labels*. It is motivated by the potential use of quantitative intuitions about how the data corresponds to labels. To restate a familiar example, a clinician might say, "when a patient is over 70 years old and has had a heart rate over 125 beats for over a minute such an alarm is pretty likely of being high priority". A weak label is defined as $\hat{y}_i = \mathcal{C}^w(x_i)$, where $\mathcal{C}^w$ is a weak labeling function. Such weak labels are a source of *weak supervision* to the learning algorithm. One often gains from generating multiple such weak labels $\{\hat{y}_i^1, \hat{y}_i^2, \ldots, \hat{y}_i^{N_w}\}$ corresponding to each input sample x_i, assuming that there are multiple experts (N_w) who are willing to suggest such weak labels. The next step is to automatically combine such weak labels and therefore help achieve better labels for an unlabeled sample x_i. This is known as data-programming [7] in the literature. Such weak labels sourced from multiple experts can be combined by tools such as *Snorkel* to suggest a single probability distribution and therefore a single possible label corresponding to each unlabeled sample.

Data programming typically advocates for the use of weak-labeling functions $\mathcal{C}^w$, which are produced by an expert, to automatically generate a weak label $\hat{y}_i^j$ for an input sample x_i. However, the challenge in recent years has been to come up with such weak labeling functions. Especially for high dimensional data like long-horizon time series signals of ICU patient vitals or diagnostic images for different pathological conditions. For instance, a weak labeling function for an image of a skin lesion would mean coming up with an algorithm which can nearly classify images to different categories in $\mathcal{Y}$. This is generally considered a hard problem, especially in the absence of large quantities of labeled samples. But, addressing the weak label generation problem can enable us to use the remaining data-programming pipeline, and build better machine learning models for smart health.

Before we state our problem, we make the following two assumptions: 1. We have access to a distance function $d : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, which can capture the degree of similarity between input pairs; and 2. An expert has the ability to label a small subset ($< N_L$) of samples with the ground truth. We can now state our research question :

Given a set of unlabeled samples $\mathcal{D}_u$, a distance function d, and an ability to label at most N_L samples, is it possible to automatically generate $N_w = \left\lfloor \frac{N_L}{|\mathcal{Y}|} \right\rfloor$ sets of weak labels for the unlabeled samples in $\mathcal{D}_u$?

We will answer this question in detail over the next few sections. In Section IV we give a brief overview of the steps involved at a high level. In Sections V-B and V-C we go over the details of the different distance functions we used for two different data types in medicine. Finally, in Section VI, we demonstrate the effectiveness of our method.

IV. Overall Approach

We outline our overall approach in this section as shown in Figure 1. To restate, we begin with an unlabelled set of samples $\mathcal{D}_u$ and a distance function d which captures the degree of similarity. Additionally, we have an expert which can give us access to ground truth up to a budget of N_L samples. The main essence of our algorithm is that we carefully figure out *which* data-samples need to be labeled in order to induce a labeling on the remaining samples. We outline the overall algorithm next:

1) Learn representative samples called memories from $\mathcal{D}_u$, with seed s^k. Let this set of memories be $M^k \subset \mathcal{D}_u$, such that $M^k := \{m_1, m_2, \ldots, m_{|M^k|}\}$.
2) This partitions the unlabeled dataset into $|M^k|$ different groups $\{g_1, \ldots, g_q, \ldots, g_{|M^k|}\}$ such that $\mathcal{D}_u = \bigcup_{q=1}^{|M_k|} g_q$ according to a nearest neighbor sense of partitioning, that is, $x_i \in g_q$ iff $q = \underset{1 \le k \le |M^k|}{\arg\min}\ d(x_i, m_k)$. This is termed as the *Memory Generation* step in Figure 1.
3) Query a clinician to label all samples in M_k. That is, $\forall m_q \in M_k$, we obtain a $y_q = C(m_q)$. This induces a weak label on all the elements in group g_q. Essentially, $\forall x_j \in g_q$ we assign a weak label $\hat{y}_j^{(k)} = C^w(x_j) = y_q$. Drawing intuitions from Figure 1, choosing a color (labels) on the memory points induces a color on all the samples in a partition.
4) Repeat steps 1 to 3 for N_w different seeds $s_1, \ldots, s_k, \ldots s_{N_w}$.
5) The weak labels according to the different seeds are inputs to the data-programming tool which creates the resulting labels. Each seed s_k acts as a distinct labeling

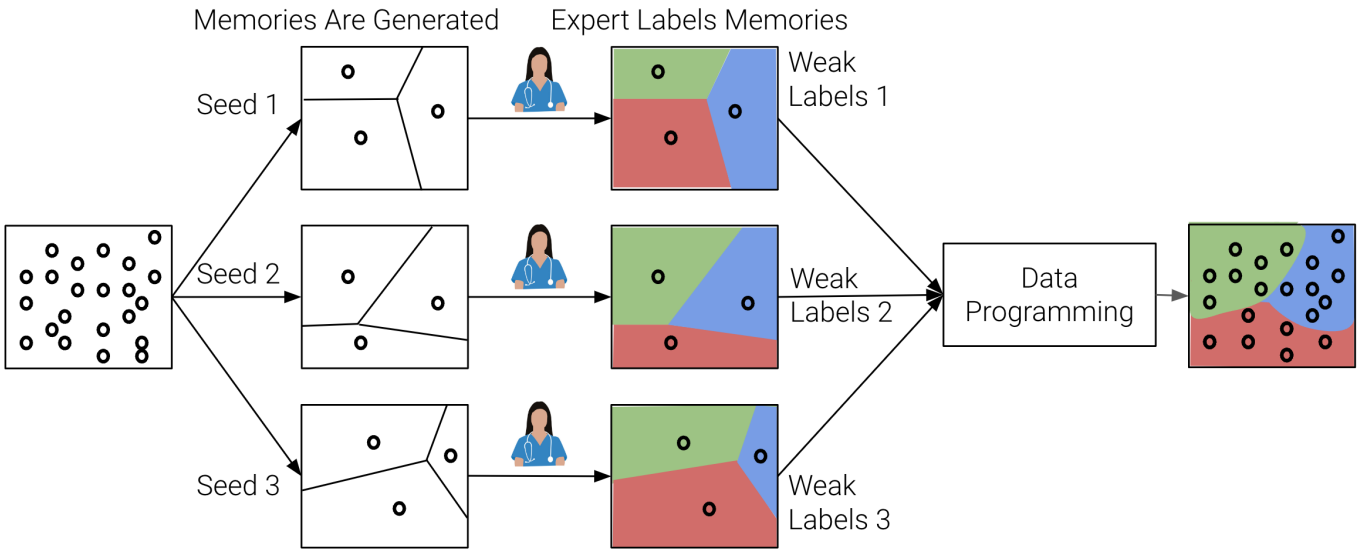


Fig. 1: Overall Approach: Starting from a dataset of unlabeled samples, we generate different partitions using different seeds. These partitions are centered around *real* prototypical samples from the dataset referred to as *memories*. Next, an expert clinician assigns a label to each prototype. This induces labels on the full dataset. Finally the the memory-induced sets of weak labels are combined using a data-programming tool to arrive at better labels.

function, producing outputs $\{\hat{y}_j^{(k)}\}$. Data programming combines the outputs of these N_w labeling functions for each sample $\left\{\hat{y}_j^{(1)}, \ldots, \hat{y}_j^{(k)}, \ldots, \hat{y}_j^{(N_w)}\right\}$ into a single probabilistic label $[p_1, \ldots, p_{|\mathcal{Y}|}]$ where $\sum_{i=1}^{|\mathcal{Y}|} p_i = 1$.

We discuss the details of the memory generation algorithm in Section V-A. However, a pertinent detail to note here is that the number of memories picked in Step 1 in the set M^k depends on a hyper-parameter called the distance threshold t. This serves as a parameter which indirectly controls the number of memory groups we end up with. We pick a threshold t such that, there are at least $|\mathcal{Y}|$ memories in M^k, for all the seeds k. That is,

$$|M^k| \geq |\mathcal{Y}|.$$

This is because if $|M^k| < |\mathcal{Y}|$, then it is guaranteed that there is at least one sample in $\mathcal{D}_u$ which is wrongly labeled. Assuming $\mathcal{D}_u$ has at least one sample from each class. Let the total number of samples to be labeled by the expert be N_s. Then,

$$N_s = \sum_{1 \leq k \leq N_w} |M^k|.$$

Therefore, the following serves as a lower bound on N_s:

$$N_s \geq N_w \times |\mathcal{Y}|.$$

Since, the maximum labeling budget is N_L, then,

$$N_L \geq N_s \geq N_w \times |\mathcal{Y}|.$$

Therefore the number of weak labels is $N_w \leq \left\lfloor \frac{N_L}{|\mathcal{Y}|} \right\rfloor$. Thus, in the worst case (performance wise) the expert is expected to label at least $|\mathcal{Y}|$ samples and the lower bound on N_L grows in proportion to the number of weak label sets.

V. Methodology

A. Memory Generation Algorithm

Unsupervised clustering, from simple k-means like algorithms to self-organizing maps (SOM) and complex neural gas algorithms [25]–[29], is a promising way to capture a distribution of the dataset. In a fashion similar to *k-means* clustering, we wish to form partitions of the data into distinct groups or clusters. Clustering with *k-means* is a well-known tool but has its challenges when used in the context of images. An issue with *k-means* is that it can potentially produce virtual cluster centers which are absent in the original dataset. This is essentially because a simple mean of two (or more) data points (for instance images in our case) might not correspond to a real data point. This is crucial since we wish to use these centers and query an expert for the labels. A sure shot to ensure *realistic* data samples is to pick from the dataset itself. Another potential issue with vanilla k-means is that it is often susceptible to outliers in the data. Hence, we restrict ourselves to partitioning around points from the unlabeled set $\mathcal{D}_u$.

The closest algorithm which achieves this is PAM [30], which is short for *Partitioning Around Medoids*. Intuitively the algorithm tries to search for centrally located data samples called *medoids* which are used to define the cluster boundaries in a nearest medoid sense. For the distance function $d : (x_1, x_2) \mapsto \mathbb{R}$ on the data set $\mathcal{D}_u$, PAM tries to select a set of r medoids - $M_r : \{m_1, m_2, \ldots, m_r\}$ such that the

Algorithm 1 Generate Memories

Input: $\mathcal{D}_u : \{x_1, x_2, \ldots x_{N_u}\}$
Output: Memories $\mathcal{M} : \{m_1, m_2, m_3, \ldots, m_r\}$
Parameter : (Max Global Steps : Z_g , Max Local Steps : Z_l, Distance Threshold t, Random Seed s)

1: BestCost = ∞
2: **for** $1 \leq g \leq Z_g$ **do**
3: Memory Set M = GenerateInitialMemories($\mathcal{D}_u, t, s, g$)
4: CurrentCost = ComputeCost(M)
5: **for** $1 \leq l \leq Z_l$ **do**
6: Perturb the memories in M locally to generate M'
7: NewCost $\leftarrow$ ComputeCost(M')
8: **if** NewCost $<$ CurrentCost **then**
9: $M \leftarrow M'$
10: CurrentCost $\leftarrow$ NewCost
11: **end if**
12: **end for**
13: **if** CurrentCost $<$ BestCost **then**
14: BestCost = CurrentCost
15: $\mathcal{M}$ = M {Store the best memory set observed}
16: **end if**
17: **end for**
18: **return** $\mathcal{M}$

following cost is minimized:

$$Cost(M_r) = \sum_{i=1}^{n} \min_{m_j \in M_r} d(m_j, x_i). \quad (1)$$

We assume that the inner minimization is always possible, and we are able to break ties arbitrarily among distinct members of the set $\mathcal{S}$. The challenge with PAM is that the naive implementation has a runtime complexity of $O(N_u^2 r^2)$ [31]. Even though there exists faster variants, they are still largely inaccessible for applications at the scale of image data sets generated in medical settings.

In order to circumvent this challenge we use a variant of the *Clustering Large Applications based upon Randomized Search* (CLARANS) [32] algorithm introduced in [33], [34]. For reference we outline this process in Algorithm 1, which combines randomized global search with a local clustering method. The algorithm aims to minimize the objective in Equation 1 and returns a subset of the training points. These are referred to as *memories* from here on. We discuss the working of this algorithm in further detail next.

Algorithm to Generate Memories: The algorithm starts with a reasonable choice for initial memories M using $GenerateInitialMemories$. The aim of this function is to ensure that each sample in $\mathcal{D}_u$ is within a distance of at most t from a memory in M. These memories are randomly chosen according to the seed s, but ensure coverage of all points. Notice that we do not choose the number of memories a priori but instead gets picked as a consequence of *distance score* d. Next, starting from this initial choice, the inner loop (Lines $5-12$) greedily looks for local improvements for a fixed number of iterations Z_l. The local improvements involve switching a memory for a nearby data sample as a candidate replacement. The partitioning cost for the choice of memories is computed by the function $ComputeCost$ which evaluates Equation 1. The outer loop of the algorithm (Lines $2-17$) resets the initial choice repeatedly for different iterations (Line 3) and keeps track of the memory set $\mathcal{M}$ with the minimum cost for each such reset. This set $\mathcal{M}$ is returned in the end. Algorithm 1 trivially terminates since each search proceeds for a fixed number of steps.

Thus, depending on the type of data, Algorithm 1 should use different distance functions d to choose the relevant memories. This is what we discuss next, where depending on the domain we choose the appropriate distance function. This is discussed in detail for time-series and image data in Section V-B and V-C, respectively.

B. Distance Metric for Medical Time Series Data

In this case study, our objective is to label physiologic monitoring alarms with respect to their suppressibility. Physiologic monitors that continuously measure parameters like blood oxygen, heart rate, and respiratory rate often overwhelm clinicians with a large amount of false alarms causing alarm fatigue [13]. Ideally, clinicians should only be alerted by the alarms that provide informative or actionable insights (i.e., *non-suppressible* alarms), while the rest of the alarms are silenced (i.e., *suppressible* alarms). Achieving this requires an alarm suppression system capable of determining the suppressibility of a potential alarm based on the patient's vitals (time series) at the time of the alarm. However, the system must be trained and evaluated prior to deployment which requires labeled time series alarm data. Traditionally, acquiring such data involves an observational study followed by manual labeling, which is expensive and time-consuming. Hence we aim to apply data programming to overcome this challenge. Unfortunately, defining labeling functions for medical time series data is generally challenging.

Formulating quantitative labeling functions can be especially challenging for clinicians, given the dynamic and intricate nature of physiological data. For instance, a clinician might describe the vitals time series as "jagged" or "erratic", which are more of a qualitative observation than a quantifiable metric. Moreover, medical time series data is inherently dynamic and can exhibit complex patterns. Designing labeling functions that capture the nuances of these patterns and their clinical significance can be a complex task. One common approach involves extracting features from a time series and defining labeling functions over them, but these primitives are not often well-defined. In contrast, devising a distance metric provides a more tractable path, leveraging mathematical principles to objectively quantify relationships between data points.

(a) Neck and/or Shoulders (b) Back

(c) Hand (d) Leg

Fig. 2: Medical image data examples. Sample dermatological images taken by patients for teledermatology consultation.

We leverage the Dynamic Time Warping (DTW) distance [35] to assess the similarity between medical time series data. DTW proves to be a robust choice for comparing medical time-series data due to its ability to handle variable speeds and temporal misalignment inherent in physiological signals. Unlike traditional metrics that assume a fixed alignment between sequences, DTW allows for flexible time warping, making it well-suited for capturing the complex dynamics of vital sign data where patients may exhibit variations in the timing of physiological events.

C. Distance Metric for Medical Image Data

In this case study, we address the challenge of identifying specific body parts in dermatological images captured by patients via smartphone cameras. Advances in smartphone camera quality have played a significant role in improving the reliability in the diagnosis of dermatological conditions through teledermatology [36]. There is considerable interest in teledermatology to develop algorithms to assist with the clinical workflow and improve efficiency. However, developing such algorithms generally requires a large and diverse annotated dataset for training, which can be time-consuming and costly to obtain due to the need for expert annotations. For example, the location of lesions on the body helps narrow a diagnosis, as many skin conditions are specific to certain body areas. Annotating body parts for a large dataset of dermatological images, however, can be tedious and challenging, particularly when images are captured by patients in uncontrolled environments (e.g., home, car, etc.) [37]. Data programming is a promising approach for overcoming these challenges associated with manual annotation of body parts in dermatological images, but defining labeling functions for medical image data also presents its own challenges.

In the context of high-dimensional data like images, defining and applying labeling functions that capture human visual understanding across different kinds of images is very challenging. In contrast, distance metrics provide a straightforward way to quantify similarity between images based on their feature representations and can be easily scaled to large image datasets. Distance functions also reduce human subjectivity in the labeling process, leading to a more consistent and objective analysis. Such distance functions can be leveraged by our approach to automatically generate labeling functions for image data, consequently facilitating an easier application of data programming.

Here, we employed the Contrastive Language–Image Pre-training (CLIP) [38] model to extract two types of features for our analysis: image representations and probability distributions across image labels. CLIP, a foundation model trained on an internet-scale dataset of images and text, is able to generalize to diverse datasets and effectively identify image-text similarities. We harness CLIP to generate the probability distribution for each image in our dataset against the 10 predefined body part labels. In addition, we used CLIP's image encoder to extract intermediate image representations with a vector length of 512.

To evaluate the similarity of the predicted probability distributions over labels, we applied the Kullback-Leibler (KL) divergence, a method for measuring the distance between two probability distributions. For the image representations, we used the Euclidean distance metric to assess the visual similarity between images.

VI. Results

In this section, we evaluate our approach on medical time series and medical image case studies.

A. Implementation

For data programming, we use the implementation of Snorkel and majority vote provided in the Snorkel Python library, version 0.9.7 (www.snorkel.org). To calculate DTW distance, we use the Python library `DTAIDistance` [39].

B. Results for Medical Time Series Data

1) Dataset: We evaluate our approach on a low oxygen saturation (i.e., SpO_2 low) alarm dataset extracted from a deidentified dataset originally collected as part of a study approved by the Institutional Review Board of a hospital. Researchers video-recorded 551 hours of patient care on a medical unit at a major US hospital during July 2014 to November

2015 from 100 children whose families and nurses provided consent. In addition, the following data was collected: patient background information, all physiologic monitoring alarms with corresponding timestamps, and continuously recorded vital signs.

To obtain the dataset of SpO_2 low alarms, we first identify the maximum length of the SpO_2 low alarms. For each alarm, we then extract the SpO_2 time series within a window of that determined length, positioning the alarm in the middle of the window. In total, 3,265 SpO_2 low alarms were extracted as part of this dataset.

Each alarm is annotated in terms of technical/clinical, valid/invalid, and actionable/non-actionable alarms [3]. We interpret these annotations with respect to suppressibility as follows: technical, valid non-actionable, and invalid alarms are interpreted as suppressible, whereas only valid actionable alarms are non-suppressible.

2) Baseline: We compare labels output by our approach to labels obtained by applying data programming with weak labels produced by domain-expert defined labeling functions. We use the set of sixty-two clinician-designed labeling functions developed for this example from [40], [41]. The labeling functions analyze the time series data to make predictions on suppressibility, *e.g.,* an alarm is non-suppressible if the heart rate is above 220 for longer than 10 seconds after the alarm starts, otherwise it abstains.

3) Result: Table I shows the accuracy and F1 of labels produced by our approach with the number of labeled samples (N_L) of 54 and 134 samples compared to that of the baseline using clinician-defined labeling functions. To produce these results, we set the max global steps Z_g to 5 and the max local steps Z_l to 30, and the distance threshold t to 90 and 60 to yield $N_L = 54$ and $N_L = 134$ samples to be labeled by the clinician, respectively. The results demonstrate that our approach yields greater accuracy and F1 scores than the baseline approach. When using majority vote we observe an improvement of approximately 17% in accuracy and 13% in F1 using our approach. Using Snorkel, we observe improvement of up to 28% on both accuracy and F1 using our approach. The results in Table I demonstrate that leveraging a small number of labeled examples supplied by a clinician can yield more accurate labels.

C. Results for Medical Image Data

1) Dataset: The dataset used in our study consists of 10,140 de-identified images, retrospectively collected from a teledermatology database. These images were captured by patients using mobile devices and uploaded into their electronic medical records at a major US hospital. For efficient labeling, we utilized the LabelMe [42] interface, adapted to identify specific body parts. Labels were based on clinical interest, covering a range of body parts such as the 'Face', 'Scalp', 'Ears', 'Torso', 'Torso-Chest', 'Torso-Abdomen', 'Breasts', 'Back', 'Neck & Shoulders', 'Arms', 'Armpits', 'Hands', 'Legs', 'Feet', 'Genitalia', 'Buttocks', 'Skin-only', and 'Not Derm' for images that were not related to dermatology. This process was focused on regions of interest (ROI) within each image.

The labeling was conducted by two annotators, with any discrepancies resolved through consensus or by marking the image as unclear, ensuring no ambiguities in body part identification remained. During the labeling phase, images with multiple ROIs had all recognizable body parts labeled. However, due to the complexities associated with multi-label classification in our label generation algorithm, we later excluded these multi-labeled images from the dataset. We also encountered images with no labels, either due to unrecognizable locations or the presence of rare body parts not defined by the labels, such as the inside of the mouth. These images were also dropped from the dataset. In addition, we merged 'Torso-Abdomen' and 'Torso-Chest' images into a single 'Torso' label. We removed 'Skin-only' and 'Not derm' labels as they were not related to body detection. Furthermore, 'Ears', 'Breasts', 'Genitalia', and 'Buttocks' were discarded due to their low frequencies, each having less than 100 images. After refining the dataset, 6,293 images with single labels were retained for analysis. The distribution of these images across the various body parts is shown in Table II.

2) Baseline: We use Snuba [43] as a baseline method for our experiment. Snuba is a system designed to overcome the challenges of gathering high-quality training labels for deep learning models. Snuba automatically generates the heuristics that labels a small subset of the data for which it is most accurate and applies this process iteratively to a large unlabeled dataset.

3) Result: We transform our multi-class dataset into ten one-versus-all labeling problems and apply our approach to each class individually. This setup enables a direct comparison with the Snuba baseline, as Snuba's implementation currently only supports a binary setting. We report the accuracy and F1 of the labels produced by our approach using CLIP image representations and probability distributions for varying labeling budgets N_L in Table V. To produce the results for image representations, we set the max global steps Z_g to 5 and the max local steps Z_l to 30, and the distance threshold t to 0.875 and 0.825 to yield $N_L = 59$ and $N_L = 113$ samples to be labeled by the clinician, respectively. For probability distributions, we set the max global steps Z_g to 5 and the max local steps Z_l to 30, and the distance threshold t to 0.8 and 0.5 to yield $N_L = 54$ and $N_L = 116$, respectively. We summarize our results by computing the average accuracy and average F1 score across all ten one-versus-all labeling problems, which are presented in Table III and Table IV. The results show our approach, using both majority vote and Snorkel, generally demonstrates improvement over Snuba. Using image representation features, our approach yields improved accuracy over Snuba (up to 6%). Unfortunately, we do not observe consistent improvement in F1 score over Snuba, but the F1 scores are generally comparable. In contrast, for probability distribution features, our approach consistently outperforms Snuba in both accuracy and F1. We observe a significant increase in accuracy by 5-15% and in F1 score by 12-19%.

Metric	Clinician-Defined LFs		Our Approach			
			$N_L = 54$		$N_L = 134$	
	Majority Vote	Snorkel	Majority Vote	Snorkel	Majority Vote	Snorkel
Accuracy	0.630	0.467	<u>0.807</u>	0.746	**0.808**	0.642
F1	0.734	0.536	<u>0.872</u>	0.823	**0.873**	0.728

TABLE I: Performance of labels generated by our approach with varying labeling ability N_L compared to labels from data programming with weak labels from clinician-defined labeling functions. We highlight the best performance scores across approaches in bold, and underline the second best scores.

Classes	Number of Images
Face	937
Scalp	998
Torso	299
Back	381
Neck and/or Shoulders	352
Arms	702
Armpits	185
Hands	1,066
Legs	901
Feet	472
Total	6,293

TABLE II: Medical image dataset class distribution.

Moreover, our results show that our approach using majority vote outperforms our approach using Snorkel in terms of both accuracy and F1 scores. Given that about half of the 10 classes have a conflict rate above 10% among labeling functions, majority voting tends to align with the correct label more often than Snorkel's probabilistic model, despite these disagreements. Furthermore, when analyzing the features, the probability distribution feature using KL divergence consistently outperforms the Euclidean distance metric on image representation features. This is evident even when the number of labeled samples (N_L), determined by adjusting the distance threshold, are roughly the same as those used for image representation. We believe KL divergence effectively captures the subtle nuances in class probability distributions, whereas Euclidean distance measures the absolute differences in feature values, potentially overlooking non-linear perceptual differences.

Finally, we apply our approach to perform multi-class labeling of the dataset. Table VI shows the F1 scores of the labels produced by our approach compared to that of the top-1 predicted label output by CLIP. CLIP's performance is inconsistent, either excellent or extremely poor, achieving an overall weighted F1 score of 52.7% and an overall accuracy of 55.2%. On the other hand, our method shows more balanced distribution across classes. It performs better on classes where CLIP struggles, such as Back, Neck and/or Shoulders, and Armpits. Our approach generally yields higher overall F1 score and accuracy, with the exception of when we use majority vote with CLIP image representations and labeling budget $N_L = 59$. In this case the overall accuracy is marginally lower than CLIP's. However, our method using CLIP's probability distribution for the majority vote with $N_L = 54$ shows the best

N_L	Metric	Snuba	Our Approach	
			Majority Vote	Snorkel
59	Accuracy	0.883 (0.019)	**0.930** (0.032)	<u>0.898</u> (0.066)
	F1	**0.431** (0.031)	<u>0.410</u> (0.288)	0.401 (0.268)
113	Accuracy	0.869 (0.009)	**0.931** (0.030)	<u>0.898</u> (0.048)
	F1	<u>0.454</u> (0.010)	**0.459** (0.263)	0.426 (0.246)

TABLE III: Performance comparison between weak labels generated by Snuba and our approach using **Euclidean distance** on CLIP image representation

N_L	Metric	Snuba	Our Approach	
			Majority Vote	Snorkel
54	Accuracy	0.788 (0.004)	**0.934** (0.038)	<u>0.918</u> (0.058)
	F1	0.320 (0.005)	**0.515** (0.335)	<u>0.492</u> (0.359)
116	Accuracy	0.891 (0.004)	**0.941** (0.027)	<u>0.916</u> (0.046)
	F1	0.453 (0.007)	**0.579** (0.264)	<u>0.566</u> (0.288)

TABLE IV: Performance comparison between weak labels generated by Snuba and our approach using **KL divergence** on CLIP probability distribution feature

results, with a 12.1% higher accuracy and a 14.5% higher F1 score compared to CLIP.

D. Ablation Study

In this section, we analyze the impact of the number of labeled samples on the accuracy of labels produced by our approach. Note that the number of labeled samples used captures our labeling budget (N_L) when using a clinician in the loop. The value of N_L depends on the distance threshold t, thus this ablation study directly investigates the influence of the distance threshold hyper-parameter on the performance of our approach. We plot the trend in labeling accuracy and F1 score for the time series case study in Figure 4 and the medical image case study in Figure 3. In general, our approach demonstrates consistently high labeling accuracy and growing F1 as the number of labeled samples increases. However, we note that for the time series case study in Figure 4, in particular for Snorkel, the trend in label quality is noisy. We attribute this to the inherent noise in our approach and plan to quantify its impact on our approach's performance in future work.

E. Limitations

In this section, we discuss the limitations of our approach. Recall that our method relies on pre-selected distance functions and, for image data, high-dimensional features over

Positive Class	Metric	Our Approach							
		CLIP Image Representation				CLIP Probability Distribution			
		$N_L = 59$		$N_L = 113$		$N_L = 54$		$N_L = 116$	
		Majority Vote	Snorkel	Majority Vote	Snorkel	Majority Vote	Snorkel	Majority Vote	Snorkel
Face	Accuracy	0.921	0.786	0.928	0.830	0.897	0.815	0.903	0.924
	F1	0.650	0.560	0.723	0.608	0.566	0.562	0.535	0.724
Scalp	Accuracy	0.954	0.898	0.95	0.918	0.962	0.961	0.961	0.923
	F1	0.849	0.748	0.835	0.781	0.878	0.878	0.881	0.796
Torso	Accuracy	0.936	0.944	0.953	0.952	0.960	0.960	0.947	0.852
	F1	0.233	0.102	0.007	0.000	0.302	0.302	0.477	0.336
Back	Accuracy	0.942	0.939	0.935	0.882	0.870	0.842	0.924	0.885
	F1	0.241	0.000	0.401	0.378	0.323	0.304	0.401	0.372
Neck and/or Shoulders	Accuracy	0.944	0.945	0.940	0.943	0.940	0.944	0.931	0.944
	F1	0.000	0.028	0.157	0.095	0.083	0.000	0.077	0.000
Arms	Accuracy	0.867	0.874	0.886	0.886	0.888	0.888	0.900	0.834
	F1	0.403	0.423	0.392	0.392	0.059	0.000	0.468	0.527
Armpits	Accuracy	0.971	0.971	0.967	0.967	0.925	0.884	0.965	0.938
	F1	0.000	0.000	0.221	0.221	0.349	0.296	0.454	0.403
Hands	Accuracy	0.930	0.908	0.934	0.870	0.975	0.975	0.971	0.966
	F1	0.789	0.776	0.793	0.711	0.923	0.926	0.931	0.916
Legs	Accuracy	0.875	0.767	0.866	0.818	0.945	0.935	0.931	0.916
	F1	0.338	0.535	0.498	0.484	0.801	0.789	0.756	0.757
Feet	Accuracy	0.956	0.941	0.952	0.918	0.981	0.979	0.976	0.975
	F1	0.598	0.411	0.558	0.591	0.869	0.859	0.832	0.840

TABLE V: Performance of our approach on a class-by-class basis on the medical images case study with CLIP image representations and with CLIP probability distributions.

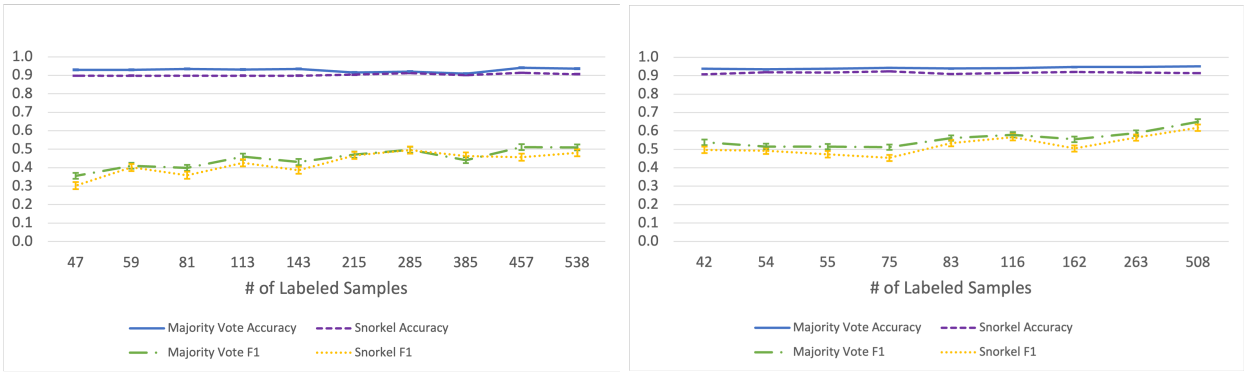
Metric	CLIP	Our Approach							
		CLIP Image Representation				CLIP Probability Distribution			
		$N_L = 59$		$N_L = 113$		$N_L = 54$		$N_L = 116$	
		Majority Vote	Snorkel	Majority Vote	Snorkel	Majority Vote	Snorkel	Majority Vote	Snorkel
Face	0.891	0.708	0.774	0.712	0.636	0.574	0.602	0.606	0.606
Scalp	0.911	0.777	0.680	0.819	0.798	0.878	0.878	0.859	0.839
Torso	0.667	0.204	0.248	0.512	1.000	0.393	0.302	0.439	0.414
Back	0.000	0.349	0.420	0.333	0.345	0.331	0.308	0.393	0.384
Neck and/or Shoulders	0.167	0.220	0.500	0.261	0.368	0.131	0.083	0.173	0.104
Arms	0.750	0.350	0.314	0.402	0.446	0.152	0.074	0.489	0.465
Armpits	0.149	0.089	0.118	0.230	0.333	0.330	0.312	0.424	0.454
Hands	0.943	0.736	0.710	0.767	0.748	0.923	0.925	0.915	0.917
Legs	0.830	0.552	0.584	0.449	0.463	0.800	0.794	0.750	0.743
Feet	0.827	0.865	0.865	0.737	0.845	0.868	0.860	0.829	0.832
Overall ACC	0.552	0.551	0.565	0.584	0.604	0.636	0.634	0.673	0.669
Weighted F1	0.527	0.552	0.540	0.578	0.576	0.633	0.619	0.672	0.661

TABLE VI: Performance comparison (F1 score) between labels generated by our approach and the top-1 label output by CLIP

which distance is computed. These items effect the performance of the memory generation algorithm which partitions the data and determines the subset of representative samples (memories) for the clinician to label. However, in certain real-world scenarios, the features and distance functions our approach uses may not be effective. The features may lack relevance for comparing samples or the distance functions may not effectively capture the similarity between samples. This can lead to a poor partitioning of the data, resulting in low accuracy weak labels when the clinician's labels for the memories are imposed on the respective partitions. Therefore, the effectiveness of our approach hinges heavily on the suitability of the pre-selected distance functions or high-dimensional features for the application.

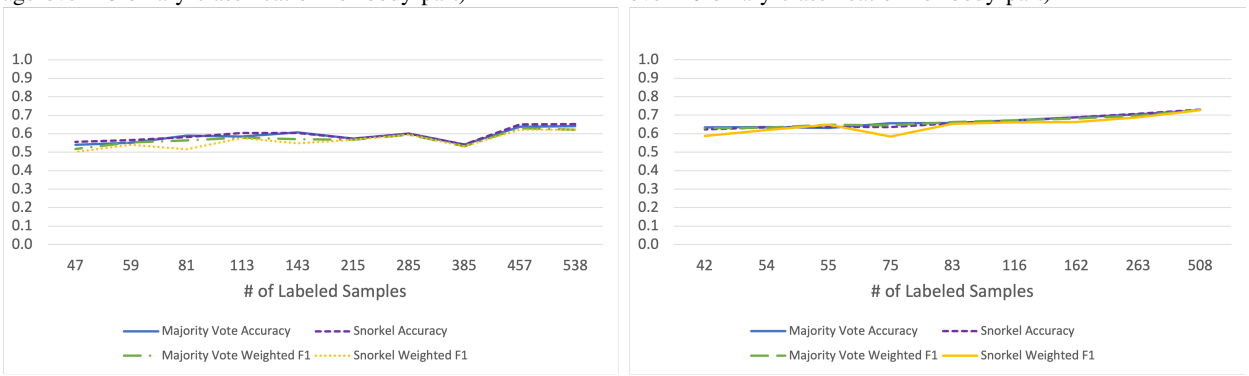
As studied in Section VI-D, the performance of our approach is impacted by the clinician's labeling budget N_L. As a clinician increases their budget, the quality of the labels produced by our approach generally improves. Consequently, poorly chosen (e.g., very small) labeling budgets can cause our approach to produce inaccurate probabilistic labels. For the evaluation in this paper, we explicitly chose labeling budgets that would be considered reasonable for an clinician in our case studies. However, in practice, determining an appropriate labeling budget may not be straightforward and may require consideration of several factors. For example, the complexity of the labeling task (e.g., the time and expense associated with annotating a data sample) is an important factor for an clinician with time constraints. Additionally, a clinician may have a desired tolerance for incorrect labels (e.g., a threshold on false positive labels), thus posing a trade-off where a larger labeling budget may be necessary to ensure higher label quality. We plan to explore these factors and devise guidelines based on them for selecting the labeling budget N_L in future work.

Lastly, our approach requires the clinician to manually



(a) Euclidean distance using CLIP image representation (average over 10 binary classification for body part)

(b) KL divergence using CLIP probability distribution (average over 10 binary classification for body part)



(c) Euclidean distance using CLIP image representation (multi-class classification)

(d) KL divergence using CLIP probability distribution (multi-class classification)

Fig. 3: F1 and Accuracy of our approach using varying number of labeled samples (N_l) for the medical image case study.

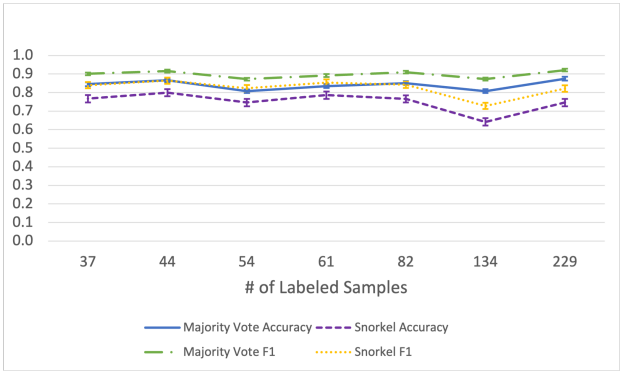


Fig. 4: F1 and Accuracy of labels output by our approach using varying number of labeled samples (N_L) for the medical time series case study.

label a subset of the data determined by the memory generation algorithm, with the subset's size being bounded by the clinician's labeling budget. As mentioned previously, manual labeling can be both time-consuming an expensive. Therefore, despite being a small portion of the data, this requirement can still pose a limitation in certain real-world scenarios. In the medical time series data case study, approximately 54 seconds was spent labeling each alarm in the dataset, roughly 55 hours to label all 3,625 alarms [12]. Using our approach with a labeling budget of 134 alarms, a clinician would spend only 2 hours labeling. In the medical images case study, we demonstrate that our approach can produce high quality labels and only required an expert to label approximately 100 images instead of thousands, reducing the number of samples to be manually labeled by roughly 99%. Hence, in both case studies presented in this paper, our approach demonstrates its capability to significantly reduce manual labeling efforts while still achieving high-quality labels.

VII. Conclusion

Labeling function generation is a challenging problem in the data-programming paradigm. An assumption made when automatically coming up with weak-labeling functions is that a small labeled set is available. In this paper we show that carefully selecting *which* samples to label and querying an expert clinician post facto can have a significant impact on performance. Additionally when compared to similar approaches this method does not need any classifier training to construct the weak labeling function from the labeled samples. In the future our plan is to study how the memory generation algorithm optimization hyper-parameters impact the accuracy of labels produced by our approach. Furthermore, we plan to reduce the number of labeled samples required, thereby

reducing the load on the clinician. We believe that leveraging large language models and existing medical guidelines may enable such an improvement.

ACKNOWLEDGMENT

This work was supported in part by NSF-1915398 and ARO MURI W911NF-20-1-0080.

REFERENCES

[1] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *Journal of The Royal Society Interface*, vol. 15, no. 141, p. 20170387, 2018.

[2] F. Piccialli, V. Di Somma, F. Giampaolo, S. Cuomo, and G. Fortino, "A survey on deep learning in medicine: Why, how and when?" *Information Fusion*, vol. 66, pp. 111–137, 2021.

[3] F. Wang, L. P. Casalino, and D. Khullar, "Deep learning in medicine—promise, progress, and challenges," *JAMA internal medicine*, vol. 179, no. 3, pp. 293–294, 2019.

[4] M. Nagendran, Y. Chen, C. A. Lovejoy, A. C. Gordon, M. Komorowski, H. Harvey, E. J. Topol, J. P. Ioannidis, G. S. Collins, and M. Maruthappu, "Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies," *bmj*, vol. 368, 2020.

[5] M. J. Willemink, W. A. Koszek, C. Hardell, J. Wu, D. Fleischmann, H. Harvey, L. R. Folio, R. M. Summers, D. L. Rubin, and M. P. Lungren, "Preparing medical imaging data for machine learning," *Radiology*, vol. 295, no. 1, pp. 4–15, 2020.

[6] P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol, "Ai in health and medicine," *Nature medicine*, vol. 28, no. 1, pp. 31–38, 2022.

[7] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré, "Data programming: Creating large training sets, quickly," *Advances in neural information processing systems*, vol. 29, 2016.

[8] D. Karimi, H. Dou, S. K. Warfield, and A. Gholipour, "Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis," *Medical image analysis*, vol. 65, p. 101759, 2020.

[9] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," *The VLDB Journal*, vol. 29, no. 2-3, pp. 709–730, 2020.

[10] J. A. Dunnmon, A. J. Ratner, K. Saab, N. Khandwala, M. Markert, H. Sagreiya, R. Goldman, C. Lee-Messer, M. P. Lungren, D. L. Rubin *et al.*, "Cross-modal data programming enables rapid medical machine learning," *Patterns*, vol. 1, no. 2, 2020.

[11] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, "Snorkel: Rapid training data creation with weak supervision," in *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, vol. 11, no. 3. NIH Public Access, 2017, p. 269.

[12] M. MacMurchy, S. Stemler, M. Zander, and C. Bonafide, "Research: Acceptability, feasibility, and cost of using video to evaluate alarm fatigue," pp. 25–33, 2017.

[13] C. W. Paine, V. V. Goel, E. Ely, C. D. Stave, S. Stemler, M. Zander, and C. P. Bonafide, "Systematic review of physiologic monitor alarm characteristics and pragmatic interventions to reduce alarm frequency," *Journal of hospital medicine*, vol. 11, no. 2, pp. 136–144, 2016.

[14] S. Evensen, C. Ge, D. Choi, and Ç. Demiralp, "Data programming by demonstration: A framework for interactively learning labeling functions," *arXiv preprint arXiv:2009.01444*, 2020.

[15] B. Denham, E. M. Lai, R. Sinha, and M. A. Naeem, "Witan: unsupervised labelling function generation for assisted data programming," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2334–2347, 2022.

[16] A. Ratner, B. Hancock, J. Dunnmon, F. Sala, S. Pandey, and C. Ré, "Training complex models with multi-task weak supervision," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4763–4771.

[17] P. Varma, F. Sala, A. He, A. Ratner, and C. Ré, "Learning dependency structures for weak supervision models," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6418–6427.

[18] W. Ren, Y. Li, H. Su, D. Kartchner, C. Mitchell, and C. Zhang, "Denoising multi-source weak supervision for neural text classification," *arXiv preprint arXiv:2010.04582*, 2020.

[19] O. Lan, X. Huang, B. Y. Lin, H. Jiang, L. Liu, and X. Ren, "Learning to contextually aggregate multi-source supervision for sequence labeling," *arXiv preprint arXiv:1910.04289*, 2019.

[20] D. Fu, M. Chen, F. Sala, S. Hooper, K. Fatahalian, and C. Ré, "Fast and three-rious: Speeding up weak supervision with triplet methods," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3280–3291.

[21] P. Varma and C. Ré, "Snuba: Automating weak supervision to label training data," in *Int. Conference on Very Large Data Bases*, vol. 12, no. 3, 2018, p. 223.

[22] N. Das, S. Chaba, R. Wu, S. Gandhi, D. H. Chau, and X. Chu, "Goggles: Automatic image labeling with affinity coding," in *2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 1717–1732.

[23] J. Li, H. Ding, J. Shang, J. McAuley, and Z. Feng, "Weakly supervised named entity tagging with learnable logical rules," *arXiv preprint arXiv:2107.02282*, 2021.

[24] X. Zhao, H. Ding, and Z. Feng, "Glara: Graph-based labeling rule augmentation for weakly supervised named entity recognition," *arXiv preprint arXiv:2104.06230*, 2021.

[25] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[26] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[27] T. Martinetz and K. Schulten, "A "neural-gas" network learns topologies," *Artificial neural networks*, vol. 1, pp. 397–402, 01 1991.

[28] K. Sridhar, S. Dutta, D. Jayaraman, J. Weimer, and I. Lee, "Memory-consistent neural networks for imitation learning," *arXiv preprint arXiv:2310.06171*, 2023.

[29] K. Sridhar, S. Dutta, J. Weimer, and I. Lee, "Guaranteed conformance of neurosymbolic models to natural constraints," in *Learning for Dynamics and Control Conference*. PMLR, 2023, pp. 76–89.

[30] *Partitioning Around Medoids (Program PAM)*, 1990, ch. 2, pp. 68–125. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470316801.ch2

[31] E. Schubert and P. J. Rousseeuw, "Faster k-medoids clustering: Improving the pam, clara, and CLARANS algorithms," *CoRR*, vol. abs/1810.05691, 2018. [Online]. Available: http://arxiv.org/abs/1810.05691

[32] R. Ng and J. Han, "Clarans: a method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003–1016, 2002.

[33] Y. Yang, R. Kaur, S. Dutta, and I. Lee, "Interpretable detection of distribution shifts in learning enabled cyber-physical systems," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, 2022, pp. 225–235.

[34] ——, "Memory-based distribution shift detection for learning enabled cyber-physical systems with statistical guarantees," *ACM Trans. Cyber-Phys. Syst.*, feb 2024, just Accepted. [Online]. Available: https://doi.org/10.1145/3643892

[35] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, USA:, 1994, pp. 359–370.

[36] J. S. Barbieri, C. A. Nelson, W. D. James, D. J. Margolis, R. Littman-Quinn, C. L. Kovarik, and M. Rosenbach, "The reliability of teledermatology to triage inpatient dermatology consultations," *JAMA dermatology*, vol. 150, no. 4, pp. 419–424, 2014.

[37] S. Sitaru, T. Oueslati, M. C. Schielein, J. Weis, R. Kaczmarczyk, D. Rueckert, T. Biedermann, and A. Zink, "Automatic body part identification in real-world clinical dermatological images using machine learning," *JDDG: Journal der Deutschen Dermatologischen Gesellschaft*, 2023.

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[39] W. Meert, K. Hendrickx, T. Van Craenendonck, P. Robberechts, H. Blockeel, and J. Davis, "Dtaidistance," Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.7158824

[40] S. Pugh, I. Ruchkin, C. Bonafide, S. Demauro, O. Sokolsky, I. Lee, and J. Weimer, "Evaluating alarm classifiers with high-confidence data programming," *ACM Transactions on Computing for Healthcare*, vol. 3, no. 4, pp. 1–24, 2022.

[41] S. Pugh, I. Ruchkin, C. P. Bonafide, S. B. DeMauro, O. Sokolsky, I. Lee, and J. Weimer, "High-confidence data programming for evaluating suppression of physiological alarms," in *2021 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 2021, pp. 70–81.

[42] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, 05 2008.

[43] P. Varma and C. Ré, "Snuba: Automating weak supervision to label training data," *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, vol. 12 3, pp. 223–236, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:54031871