Curating Naturally Adversarial Datasets for Learning-Enabled Medical Cyber-Physical Systems

Sydney Pugh
University of Pennsylvania
Philadelphia, PA 19104
sfpugh@seas.upenn.edu

Ivan Ruchkin
University of Florida
Gainesville, FL 32611
iruchkin@ece.ufl.edu

James Weimer
Vanderbilt University
Nashville, TN 37235
james.weimer@vanderbilt.edu

Insup Lee
University of Pennsylvania
Philadelphia, PA 19104
lee@seas.upenn.edu

***Abstract*—In medical cyber-physical systems (CPS), where patient safety is a top priority, the robustness of learning-enabled components (LECs) becomes crucial. Therefore, a comprehensive robustness evaluation is necessary for the successful deployment of these systems. Existing research predominantly focuses on robustness to *synthetic adversarial examples*, crafted by adding imperceptible perturbations to clean input data. However, these synthetic adversarial examples do not accurately reflect the most challenging real-world scenarios, especially in the context of healthcare data. Consequently, robustness to synthetic adversarial examples may not necessarily translate to robustness against *naturally occurring adversarial examples*. We propose a method to curate datasets comprised of natural adversarial examples to evaluate the robustness of LECs. The method relies on probabilistic labels obtained from automated weakly-supervised labeling that combines noisy and cheap-to-obtain labeling heuristics. Based on these labels, our method adversarially orders the input data and uses this ordering to construct a sequence of increasingly adversarial datasets. Our evaluation on six medical CPS case studies and three non-medical case studies demonstrates the efficacy and statistical validity of our approach to generating naturally adversarial datasets.**

I. Introduction

Deep learning models have demonstrated remarkable performance in the analysis of medical time-series data [1], [2], inciting substantial interest in the development of learning-enabled medical cyber-physical systems (CPS) [3], [4]. For example, physiologic monitoring systems are an essential CPS for healthcare that continuously measure patient vitals and raise alarms when the vitals appear abnormal. Unfortunately, such systems are known to overwhelm caregivers with many inactionable or non-informative alarms [5], [6]. Recent research have proposed novel learning-based algorithms for monitoring and suppressing unnecessary alarms [7], [8], [9]. Thus a plausible next step would be to integrate such learning-based alarm suppression algorithms into a physiologic monitoring system. However, patient safety is a primary concern for medical CPS and thus the learning-enabled components must function properly on expected and unexpected inputs [10]. For instance, a learning-enabled physiologic monitoring system should be able to handle patients with abnormal physiology due to illness or instances where the sensor data is noisy due to artifact or missing due to sensor disconnections. Successful deployment of learning-enabled medical CPS, therefore, is contingent on a thorough evaluation of the system's robustness.

Robustness is typically evaluated by observing a model's performance against adversarial examples, particularly *synthetic* adversarial examples, that are generated by intentionally adding small perturbations to *labeled* input data to cause misclassification while being nearly imperceptible to humans. This evaluation approach can be challenging to apply to a time-series medical CPS for two primary reasons. Firstly, synthetic adversarial examples generally do not resemble adversarial examples that would be encountered in the real world [11] – simply adding random perturbations to medical data usually yields invalid and unrealistic examples. Secondly, it is time-consuming and expensive to construct highly accurate labeled datasets for generating realistic synthetic adversarial examples. A common way to do so is to collect data via an observational study [12] and then have domain experts manually label the data. However, such a study is a major commitment when it comes to an initial deployment of a medical CPS, in part due to the significant effort of manually labeling examples.

Instead, we focus on *natural* adversarial examples — real patient examples that are difficult to classify [13]. Natural adversarial examples capture the inherent variations and uncertainties present in real-world medical data that effectively deceive models. Therefore, evaluating the robustness of medical deep learning models against such examples becomes crucial to ensuring model robustness in the real world. We aim to identify natural adversarial examples in *unlabeled* medical time-series data so that we can curate naturally adversarial *datasets*.

In an ideal world, in medical CPS, a model's accuracy would be evaluated on a labeled natural adversarial dataset – in reality, developing labeled medical datasets is an expensive and time-consuming task. However, obtaining weak labels for historical datasets is a cheap and quick alternative [14], but assessing accuracy/robustness with respect to weak labels becomes challenging due to the uncertainty surrounding the correctness of the weak labels. In our experience, weak labels for adversarial examples are prone to significant inaccuracies due to the inherent difficulty in classifying these examples.

Hence, robust models tend to disagree with the weak labels of natural adversarial examples because of weak labeling inaccuracies – not model inaccuracies. To overcome this issue, our intuition is to focus on the *change* of the model's performance rather than its absolute performance. By gradually increasing the proportion and severity of adversarial examples

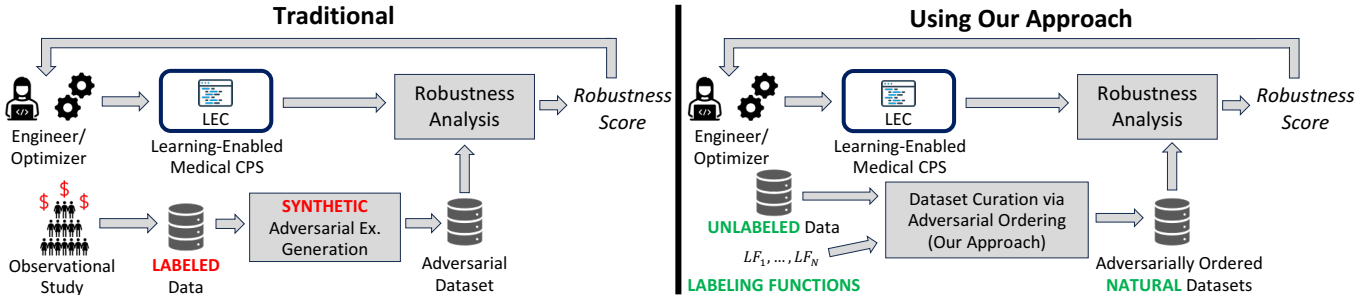

Fig. 1. Traditional robustness evaluation versus our approach. LEC = Learning-Enabled Component.

per dataset, we aim to curate an adversarially ordered sequence of datasets, which have a diminishing accuracy of weak labels. On such a sequence, a robust model would be expected to show decreasing accuracy with respect to our weak labels.

This paper introduces a weakly-supervised method for curating adversarially ordered datasets containing natural adversarial examples. Our method can support early-stage low-cost robustness evaluations of medical CPS as depicted in Figure 1. Note that our evaluation method is complementary to the traditional method as it can be applied to historical data before incurring the cost of an observational study. A key step of our method is to probabilistically label data using *weakly-supervised data labeling*. This process leverages *labeling functions*, which assign labels to subsets of the data. At a high level, these labels are combined into a single probabilistic label per example via a weighted combination, where the weights reflect the expected accuracy of the labeling functions. Labeling functions are noisy: their labels can be incorrect, incomplete, or contradictory. Furthermore, labeling functions are assumed to be conditionally independent. We start with a dataset of unlabeled time series and a set of labeling functions. We then select a subset of *independent labeling functions* to be used in our method via a heuristic. The outputs of the selected labeling functions are combined to yield a probabilistic label, i.e., a label and corresponding confidence in it, for each sample in the dataset.

Adversarial examples are by definition difficult to classify, hence the *uncertainty in the label* of an example can indicate how adversarial it is. We identify adversarial examples as those with high label uncertainty. As an indication of label uncertainty, we will use label confidence scores. Consequently, they will serve as the basis for adversarial ordering.

Unfortunately, label confidences do not reflect the true accuracies of the labels: weakly-supervised data labeling techniques are generally *poorly calibrated*. To overcome this issue, we construct *confidence intervals* for the probabilistic labels output by these techniques to quantify the uncertainty of the labeling process. An important consideration not previously considered is the *number of labeling functions* combined to produce the label. Informed by this circumstance, our intervals indicate, for a given level of confidence, the range of possible confidence values in the assigned label. Finally, using the lower bounds of confidence intervals, we order the examples by their naturally adversarial severity.

We validate our method on six medical case studies and three non-medical case studies. The medical case studies include five clinical alarm datasets derived from a 551-hour alarm labeling effort from Children's Hospital of Philadelphia (CHOP), and one medical imaging dataset extracted from the Open-i dataset (https://openi.nlm.nih.gov/faq) maintained by the National Institutes of Health (NIH). The three non-medical case studies include weather-related tweets, book reviews, and synthetic YouTube comments. Our evaluation demonstrates that our approach successfully generates natural adversarial datasets with statistically valid adversarial ordering.

In summary, this paper makes the following contributions:

- An approach to building datasets with increasing amounts of natural adversarial examples based on confidence intervals,
- A method of selecting independent labeling functions to use for weakly-supervised data labeling,
- An evaluation of the statistical validity of our curated adversarially ordered natural datasets on nine case studies.

The remainder of this paper is structured as follows. Section II discusses the literature related to this paper. Section III introduces relevant definitions and formulates our problem. Section IV describes our approach to curating adversarially ordered natural datasets, which then is evaluated in the next section, Section V. Section VI discusses the limitations and commercial value of our approach. Finally, the paper concludes with a summary in Section VII.

II. Related Work

In this section we overview related work for adversarial examples and weakly-supervised data labeling in the context of medical CPS.

A. Adversarial Examples for Medical CPS

[15] discovered the existence of *adversarial examples*: inputs intentionally designed to cause machine learning models to predict incorrectly. Subsequent research has predominately focused on *synthetic* adversarial examples, which are artificial inputs specifically generated to deceive models. A common approach for generating synthetic adversarial examples is to apply adversarial perturbations to clean inputs. For example,

ℓ_p adversarial examples are generated by perturbing an input by some worst-case distortion that is small in the ℓ_p sense [16].The small distortion is nearly imperceptible to humans, making it challenging to detect by inspection — but can have a significant negative impact on the behavior of a model. Unfortunately, ℓ_p adversarial examples are not suitable for time series medical data because adding such noise to such data typically yields invalid and unrealistic examples. Adversarial spatial transformations (e.g., rotations, translations) may also be used to perturb inputs to generate synthetic adversarial examples [17]. However, this approach is only applicable for data with spatial structure (like image data) and hence is not applicable for time-series data common in medical CPS.

An alternative approach for generating synthetic adversarial examples is to use synthetic data generation techniques. Researchers have considered using state-of-the-art patient simulators [18], [19] and Generative Adversarial Networks (GANs) [20], [21], [22] to efficiently produce perceptually realistic adversarial examples. However, both techniques have substantial limitations. Patient simulators and GANs can struggle to fully replicate complex physiological variations found in real medical data, resulting in unrealistic examples. This difficulty arises due to the inherent complexity of human physiology and the limitations of computational modeling, which require these techniques to learn simplified models of medical scenarios. Adversarial examples generated by these techniques can also be biased. The quality of the adversarial examples generated by these techniques is highly dependent on the quality of the technique's training data. Thus, biases in the training data (such as under-representation of specific ethnicities) can propagate into the generated data. In contrast to these techniques, our approach consistently produces realistic adversarial examples by sampling them from real medical data. However, our examples may reflect the bias of the input data.

Recently, [13] found that clean, realistic inputs can reliably degrade the performance of machine learning models. These inputs are referred to as *natural* adversarial examples, as they are examples that occur naturally but still to lead to erroneous model predictions. In practice, natural adversarial examples are obtained by selecting them from existing datasets. Adversarial filtration is a popular approach to select natural adversarial examples from an existing dataset by removing examples that are diverse in appearance but classified easily via very predictable classification boundaries [23]. Several works have explored filtration by removing examples solved with simple spurious cues in the image domain [13] and natural language processing (NLP) domain [24], [25], [26], [27], [28], [29]. Unfortunately, the application of this specific method may not be suitable for time-series medical data since these datasets generally lack spurious cues. However, our approach can be considered an adversarial filtration technique that removes examples based on the lower bound of weak label confidences. We leverage the intuition that low label confidence implies a sample is more "naturally adversarial". An alternative way to measure label confidence is to crowdsource, i.e., have multiple human labelers label an example and then compute the disagreement amongst the labelers. Hence, adversarial filtration via crowdsourced label uncertainty is an approach alternative to ours.

Adversarial examples typically have a different underlying data distribution than non-adversarial (clean) examples. This suggests that it may be possible to select natural adversarial examples from existing data by observing the input data distribution. For example, natural adversarial examples can present as outliers in the input data feature distribution [30]. The feature distribution can be estimated via a density estimator, and then an outlier detection method can be used to select the natural adversarial examples. Out-of-distribution (OOD) detectors [31] may also be used to identify natural adversarial examples in existing data. Adversarial examples are similar to clean examples from the training data distribution but with small imperceptible perturbations, and thus can be considered OOD. This paper focuses on selecting natural adversarial examples based on label uncertainty rather than feature values. We leave the investigation of feature-based approaches for future work.

B. Weakly-Supervised Data Labeling for Medical CPS

Recently, a quick and inexpensive way of labeling data has emerged, known as *weakly-supervised data labeling*. Often motivated by its need in medical applications, its key element is a set of quantitative intuitions about how the data corresponds to labels. For example, a clinician might say, "when a patient over 60 years old has had a heart rate over 120 beats for over a minute, such an alarm is a high priority." These intuitions, algorithmically represented as *labeling functions*, are allowed to be incomplete, sometimes incorrect, and contradictory. A labeling function returns a class label or an "abstain" verdict for any input. Given a diverse combination of many labeling functions and an unlabeled dataset, weakly-supervised data labeling techniques produce probabilistic labels for each sample in the dataset in the form of probability distributions over the label space. Such a label is represented as a probability distribution over the label space. Subsequently, for each sample, the weak label consists of the label with the highest probability and the confidence equal to that probability. A prominent weakly-supervised data labeling technique *Snorkel* [14], [32] estimates an optimal weight for each labeling function by using a generative graphical model and a prior on the class balance. Our approach takes data and labeling functions as input, feeds them into Snorkel, and builds on the resulting probabilistic labels to order the data adversarially.

Extending the above work, adversarial data programming generates data in addition to labeling it [33], [34]. A GAN is used to estimate the weight of each labeling function as well as the dependencies between them given a set of labeling functions and an unlabeled dataset. The weights and dependencies are used by the GAN's generator to produce labeled samples that come from the data distribution. Hence, one may be able to train an adversarial data programming model to generate more examples given a dataset comprised of natural adversarial examples. However, as mentioned earlier, GAN generated examples can be unrealistic and biased – especially in medical

CPS applications where small perturbations to physiological waveforms can profoundly change their meaning.

III. Adversarially Ordered Natural Datasets

We start with several notational conventions. Given a set B, we write $|B|$ to be the set's cardinality. Given a value $v \in \mathbb{R}$, we write $|v|$ to be the absolute value. We are given an evaluation dataset $X \subset \mathcal{X}$ with *unknown* true labels Y.

Before formally stating the problem considered in this work, we define adversarially ordered datasets and statistically valid adversarially ordered datasets.

Definition 1 (Adversarially Ordered Natural Datasets): Consider a sequence of natural (non-synthetic) datasets $D_1, \ldots, D_N$, where $D_i = (X_i, Y_i, \hat{Y}_i)$ is composed of samples X_i, corresponding unknown true labels Y_i, and corresponding known noisy labels $\hat{Y}_i$. These datasets are *adversarially ordered* if their accuracy (non-strictly) monotonically decreases. That is,

$$\text{ACC}(Y_1, \hat{Y}_1) \geq \text{ACC}(Y_2, \hat{Y}_2) \geq \ldots \geq \text{ACC}(Y_N, \hat{Y}_N)$$

where

$$\text{ACC}(Y, \hat{Y}) = \frac{1}{|Y|} \sum_{i=1}^{|Y|} \mathbf{1}\left(y_i = \hat{y}_i\right)$$

In short, the accuracy of the weak labels for each dataset in adversarially ordered natural datasets does not increase. It is important to validate this trend to verify its direction and significance. Spearman's rank correlation coefficient ρ is a widely used measure of the strength and direction of a monotonic relationship [35]. The coefficient ρ is a value between -1 and 1, where close to -1 indicates a strong, monotonically decreasing trend and close to 1 indicates a strong, monotonically increasing trend. The coefficient is accompanied by a p-value p^*, which quantifies the statistical significance of the observed relationship. The following definition describes how to compute Spearman's Rank Correlation for adversarially ordered natural datasets.

Definition 2 (Spearman's Rank Correlation): Consider adversarially ordered natural datasets $D_1, \ldots, D_N$. Let $R = (i_1, \ldots, i_N)$ where $i_n \in \{1, \ldots, N\}$ be the rank order of the accuracies of the datasets (i.e., $\text{ACC}(Y_{i_1}, \hat{Y}_{i_1}) \leq \text{ACC}(Y_{i_2}, \hat{Y}_{i_2}) \leq \ldots \leq \text{ACC}(Y_{i_N}, \hat{Y}_{i_N})$). We compute *Spearman's rank correlation* coefficient ρ as follows:

$$\rho = \frac{6 \sum_{j=1}^{N} (i_j - j)^2}{N(N^2 - 1)}$$

The corresponding p-value is $p^* = 2 \times \mathbf{P}(T \geq |t|)$, where T follows a t-distribution with $N - 2$ degrees of freedom and

$$t = \rho \sqrt{\frac{N-2}{1-\rho^2}}$$

We determine the statistical validity of adversarially ordered datasets by checking that Spearman's rank correlation is negative and p-value is below a predetermined significance level. We formalize this idea in the definition below.

Definition 3 (Statistically Valid Adversarial Ordering): Consider adversarially ordered natural datasets $D_1, \ldots, D_N$. Let ρ and p^* be the Spearman's Rank Correlation Coefficient and corresponding p-value, computed from the weak label accuracies of the datasets. The datasets are *statistically valid adversarially ordered* if the coefficient is negative (i.e., $\rho < 0$) and the p-value is statistically significant (i.e., $p^* \leq \gamma$ where γ is the predetermined adversarial ordering significance threshold).

We are now ready to state our central technical problem.

a) Problem: Given unlabeled data X and labeling functions Λ, produce statistically valid adversarially ordered natural datasets $D_1, \ldots, D_N$.

Now we highlight the challenges of obtaining statistically valid adversarially ordered medical datasets — and our steps to overcome them. Our main challenge is the absence of true labels in a given dataset X, which makes it impossible to directly compute the accuracies in Def. 1. To address this challenge, we will create a *probabilistic labeler* — an algorithm that takes a sample and assigns it an estimated label and a confidence in that label (a value between 0 and 1). We refer to the assigned label as $\hat{f}(x)$ and its confidence as $\hat{g}(x)$ for any sample $x \in \mathcal{X}$. We will build that labeler from *labeling functions*, which encode the rules of thumb and heuristics acquired from medical experts. A labeling function $\lambda : \mathcal{X} \to \mathcal{Y} \cup \{0\}$ takes a sample $x \in \mathcal{X}$ and either abstains (i.e., assigns label 0) or assigns one of the classes to it. Labeling functions can contradict each other or abstain in different combinations.

Our second challenge is that probabilistic labelers are generally overconfident in their estimated labels (as our experience shows). That is, confidence scores $\hat{g}(x)$ are unreliable and should not be used as an indication of the accuracy of label $\hat{f}(x)$. Hence, our second task is to better quantify the uncertainty in the estimated labels. For each sample $x \in X$, we generate an interval I of possible confidences in label $\hat{f}(x)$. This interval should contain the *true* confidence in label $\hat{f}(x)$, call it $g(x)$, with probability of at least $1 - \alpha$ where α is the significance level specified by the user. The next section details our solution to the problem of this paper.

IV. Dataset Curation via Adversarial Ordering

This section describes our approach to curate adversarially ordered natural datasets. Figure 2 summarizes the steps of our approach. Our approach takes as input an unlabeled dataset X and a set of labeling functions Λ. The first step of our approach selects a subset of the labeling functions to be used in the next step, probabilistic labeling of the unlabeled dataset. Next, we quantify the uncertainty in the probabilistic labels by constructing confidence intervals around them. Then the confidence intervals are used in the final step to curate a sequence of progressively more adversarial datasets.

A. Labeling Function Pruning

Labeling functions are a key component of weak supervision techniques, offering a pragmatic approach for annotating data with noisy yet informative labels. However, in practice, labeling functions often exhibit statistical dependencies between

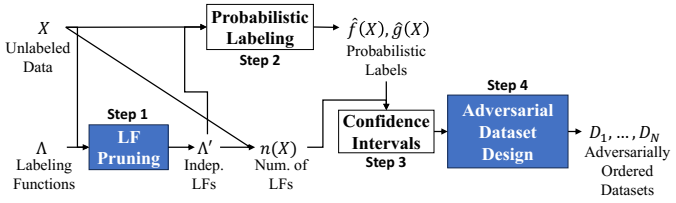

Fig. 2. Our approach for curating adversarially ordered natural datasets.

Input: Labeling functions Λ, unlabeled data X, and correlation threshold δ

Output: Independent subset of labeling functions $\Lambda' \subseteq \Lambda$

1: Apply labeling functions Λ to the data X to get weak labels $\mathcal{L}_\Lambda(X)$.
2: Compute the correlation c_{ij} between all pairs of labeling functions $\lambda_i, \lambda_j \in \Lambda$ where $i \neq j$ using $\mathcal{L}_\Lambda(X)$.
3: Construct a labeling function dependency graph G with labeling functions λ as nodes and edges between labeling functions (nodes) with correlation $c_{ij} > \delta$.
4: Rank labeling functions in descending order by the number of maximal cliques of G they belong to, breaking ties by labeling function coverage (higher coverage earns higher rank). Let R denote this ranking.
5: Let $\Lambda' = \Lambda$. For each labeling function λ in the ranking R, if $\lambda \in \Lambda'$, remove all other labeling functions with which it shares a maximal clique in G from Λ'.

Algorithm 1: Selecting independent labeling functions.

them, (e.g., multiple functions relying on the same features or patterns) [14]. Dependent labeling functions result in duplicate information that can bias the outputs of weak supervision techniques, unless accounted for in the underlying model. Most techniques commonly assume the independence of labeling functions (conditioned on the true label); however, they also offer the option to provide labeling function dependencies as additional input [14], [36]. When provided, the dependencies are embedded into the underlying model. Unfortunately, in many applications the number of dependencies can become too large for the technique, especially when given many labeling functions. Hence it may be preferred to provide a smaller set of independent labeling functions instead of a larger set of dependent ones. We opt to use weakly-supervised data labeling techniques under the independent labeling function assumption. Therefore, it is crucial to identify a subset of the labeling functions $\Lambda' \subseteq \Lambda$ where the labeling functions are independent of each other prior to applying these techniques.

The pseudo-code for our independent labeling function selection procedure is presented in Algorithm 1. First, the labeling functions Λ are applied to the dataset X to obtain the weak labels. We will refer to the weak labels for X as $\mathcal{L}_\Lambda(X) = (\mathcal{L}_{\lambda_i}(X), \ldots, \mathcal{L}_{\lambda_{|\Lambda|}}(X))$. Next, we compute the Pearson correlation coefficients c_{ij} for each pair of labeling functions (λ_i, λ_j) where $\lambda_i, \lambda_j \in \Lambda$ and $i \neq j$ from the weak labels, i.e., $c_{ij} = \textsc{Pearson}(\mathcal{L}_{\lambda_i}(X), \mathcal{L}_{\lambda_j}(X))$. These coefficients are measures of linear correlation (dependence) between pairs of our labeling functions.

Now we aim to rank the labeling functions from most to least independent. To do this, we first construct a graph representation of the labeling function dependencies [36]. Concretely, the nodes of the graph G are the labeling functions in Λ. We add an edge to G for each labeling function pair whose correlation is sufficiently large, that is, add edge (λ_i, λ_j) if $|c_{ij}| > \delta$ where δ is a user-specified minimum threshold on correlation. Next, we identify maximal cliques of labeling functions, which effectively reveal subsets of labeling functions that tend to share similar labeling patterns (i.e., cover the same information). Hence our intuition is that labeling functions belonging to many cliques are more dependent. We rank the labeling functions by the number of maximal cliques they belong to in G. Ties in the ranking are resolved by labeling function coverage, that is, the proportion of samples for which a labeling function emits a (non-abstain) label, i.e., $\frac{1}{|X|}\sum_{x \in X} \mathbf{1}\left(\lambda(x) \neq 0\right)$. We assign higher rank to labeling functions with higher coverage to retain a substantial proportion of weak labels for the data X.

Finally, we determine a subset of independent labeling functions $\Lambda' \subseteq \Lambda$. Our goal is to select the smallest subset of labeling functions that cover all the cliques. Let $\Lambda' = \Lambda$ to start. Then for each labeling function λ in the ranking (in descending order), if $\lambda \in \Lambda'$, then remove all other labeling functions that share a maximal clique with λ. Now we have an independent subset $\Lambda' \subseteq \Lambda$ for weak supervision.

B. Probabilistic Labeling

In this step, we combine the weak labels produced by the labeling functions into a single "strong" probabilistic label. This label is characterized by a confidence score between 0 and 1, indicating the level of certainty in the label's accuracy. Mathematically, for each sample $x \in X$, we combine its weak labels $\mathcal{L}_\Lambda(x)$ into a probabilistic strong label $\hat{f}(x)$ with confidence $\hat{g}(x)$.

Our approach computes the strong label and corresponding confidence using a weak supervision technique that performs a *weighted combination* over the weak labels. The weights w has a fixed vector per class containing one weight per labeling function. While our approach supports a variety of weighted combination techniques, we consider two of them in this paper:

- Majority vote
- Generative model with an uninformed prior

These two techniques are applied in two steps:

1) Determine the weights w
2) Combine the weak labels $\mathcal{L}_\Lambda(x)$ using weights w for each $x \in X$

The first step learns a non-negative weight vector where each weight indicates the relative priority of the corresponding labeling function. Majority vote, a widely-used and straightforward method for combining multiple discrete signals into one, assigns equal priority to each labeling function. Hence,

the weight vector for majority vote is uniform (i.e., $w_i^{(y)} = 1$ for all $i \in \{1, \ldots, |\Lambda|\}$ and $y \in \mathcal{Y}$). Generative models are popular in state-of-the-art weak supervision literature [14], [37], [33], [38] These models give higher weights to labeling functions with higher accuracies. The accuracies are unknown a priori, so the model estimates them by observing the agreements and disagreements of the labeling functions in the data ($\mathcal{L}_\Lambda(x)$) during its training phase. Furthermore, the model can be trained with a prior that specifies the expected frequency of each label in $\mathcal{Y}$. We assume the actual prior is unknown and we do not have a labeled dataset on which we can estimate it, so we use an uninformed prior which assigns equal probability to every label. The technical details on this use of generative models can be found in [39].

In the second step, the learned weights are used to combine the weak labels into a single probabilistic label for each sample. Suppose now we want to label a sample x given its weak labels $\mathcal{L}_\Lambda(x)$ and the weights w. First, we obtain label weights by adding up the weights of all labeling functions choosing that label, i.e., $\sum_{i=1}^{|\Lambda|} w_i^{(y)} \cdot \lambda_i(x)$ for $y \in \mathcal{Y}$. Then we pass the label weights through the softmax function, which is a standard way to normalize positive real numbers into a probability distribution. Finally, the largest normalized weight is used as the confidence $\hat{g}(x)$ and the label corresponding to that weight becomes the estimated label $\hat{f}(x)$.

Unfortunately, as mentioned in the previous section, the confidence scores output by these two weak supervision techniques are typically over-confident; that is, the confidence $\hat{g}(x)$ overestimates the accuracy of the estimated label $\hat{f}(x)$. In other words, weak supervision techniques are poorly calibrated [40] – the confidence scores do not reflect the true accuracy of the label. Majority vote can be especially prone to over-confidence because if many of the labeling functions are inaccurate, the label prediction can be incorrect with high confidence. Generative models have an advantage over majority vote because they can account for the inaccuracies of labeling functions and, thus, yield more accurate confidences. However, as we witness in practice, generative models are also often poorly calibrated. Hence, the confidences output by weak supervision techniques should be used with caution or disregarded entirely. In response, we propose using confidence intervals to account for the uncertainty in these confidences.

C. Confidence Intervals for Weak Label Accuracies

Our confidences in the estimated labels from the prior subsection can be unreliable. More precisely, the confidences in the estimated labels may not reflect the true accuracy of those estimated labels. We quantify the potential inaccuracy in the estimated labels by providing *confidence intervals*. Our confidence intervals contain the likely true confidences in the estimated labels. For each sample $x \in X$, we aim to generate an interval I containing the true confidence in the estimated label $\hat{f}(x)$, call it $g(x)$, with probability of at least $1 - \alpha$, i.e., $\mathbf{P}[g(x) \in I] \geq 1 - \alpha$.

We bound the unknown true confidences $g(X)$ in our estimated labels $\hat{f}(X)$ using the Clopper-Pearson (CP) interval $[\theta_L, \theta_U]$ where $\theta_L, \theta_U \in [0, 1]$ and $\theta_L < \theta_U$ [41]. This interval bounds the true success probability μ, constructed from a sample $s \sim B(n, \mu)$ from a binomial distribution with n trials and success probability μ, which holds with probability at least $1-\alpha$, i.e., $\mathbf{P}_{s\sim B(n,\mu)}\left[\theta_L(\alpha; n, s) \leq \mu \leq \theta_U(\alpha; n, s)\right] \geq 1-\alpha$.

Intuitively, few non-abstaining labeling functions with large weights should yield a lower confidence than many non-abstaining labeling functions with moderate weights. Hence, in addition to the labeling function weights, the *number of labeling functions* n contributing to the estimation of the labels should be considered in the calculation of confidence in $\hat{f}(x)$. While the confidences provided by probabilistic labelers $\hat{g}(X)$ do not take this into account (which can lead to their unreliability), we incorporate the number of voting labeling functions into the construction of our intervals. When a labeling function emits a (non-abstain) label for some sample x, we consider it a Bernoulli trial whose outcome is a success if that label is correct or a failure if it is incorrect. We let the number of successes s (of the n trials) be the normalized probability of label $\hat{f}(x)$ weighted by the number of non-abstaining labeling functions n. Hence by definition of the CP interval, we derive the interval $I = [\theta_L, \theta_U]$ for $g(x)$ given any sample $x \in X$ and significance level α as,

$$\theta_L(\alpha; n, s) = B\left(\frac{\alpha}{2}; s, (n - s + 1)\right)$$

$$\theta_U(\alpha; n, s) = B\left(1 - \frac{\alpha}{2}; (s + 1), (n - s)\right)$$

where

$$n(x) = \sum_{i=1}^{|\Lambda|} \mathbf{1}\left(\lambda_i(x) \neq 0\right)$$

$$s(x) = n(x) \cdot \frac{\exp\left\{\sum_{i=1}^{|\Lambda|} w_i^{\left(\hat{f}(x)\right)} \lambda_i(x)\right\}}{\sum_{y\in\mathcal{Y}} \exp\left\{\sum_{i=1}^{|\Lambda|} w_i^{(y)} \lambda_i(x)\right\}}$$

where $B(q; a, b)$ is the q-th quantile from a beta distribution with shape parameters a and b.

D. Adversarial Dataset Curation

Now we create a sequence of adversarially ordered natural datasets. Our confidence interval lower bounds indicate the smallest possible certainty in the estimated label. So as more samples with small lower bounds are included in a dataset, the more adversarial it gets. We arrange the samples in X by their confidence interval lower bound in descending order, i.e., $x_{i_1}, x_{i_2}, \ldots, x_{i_{|X|}}$ where $\theta_L(x_{i_1}) \geq \theta_L(x_{i_2}) \geq \ldots \geq \theta_L(x_{i_{|X|}})$ and $i_1, \ldots, i_{|X|} \in \{1, \ldots, |X|\}$. Then we produce a sequence of datasets $D_1, \ldots, D_N$ where each dataset D_n contains the top $\frac{i}{N}$ percent of ordered samples and their corresponding estimated labels, that is,

$$D_n = \{(x_{i_j}, \hat{f}(x_{i_j}))\} \quad \text{for } j \in \left\{1, \ldots, \frac{i \cdot |X|}{N}\right\}$$

In conclusion, we have natural datasets $D_1, \ldots, D_N$ that are progressively more adversarial.

Example	Train	Valid	Test	Num. LFs
HR Low	79	-	-	62
HR High	1315	-	-	62
RR Low	312	-	-	62
RR High	574	-	-	62
SpO_2 Low	3265	-	-	62
Crossmodal	2630	376	378	18
Crowdsourcing	187	50	50	103
Recsys	796956	8339	42191	5
Spam	1586	-	250	9

TABLE I
SUMMARY OF OUR NINE EVALUATION DATASETS.

V. RESULTS

In this section, we evaluate our approach on nine case studies (six with clinical relevance). Specifically, we summarize the nine case studies describe comparative approaches, and evaluate the adversarial ordering of natural adversarial datasets produced by our approach.

A. Case Studies

We evaluate our approach on five physiological alarm suppression case studies, one clinical text classification case study, and three non-clinical text classification case studies. These datasets are summarized in Table I.

a) Physiologic Alarm Suppression: Alarm suppression, which involves distinguishing suppressible (uninformative to clinical care) and non-suppressible physiologic-monitoring alarms, is the aim of the first five datasets. Datasets of heart rate (HR) low/high, respiratory rate (RR) low/high, and SpO_2 low alarms were extracted from [42]. Each alarm sample is represented as a multi-vital sign time series data. We use the set of sixty-two clinician-designed labeling functions developed for this example from [43], [39]. 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.

b) Clinical Text Classification: The **Crossmodal** dataset aims to label radiography images by writing labeling functions over an auxiliary modality, namely, corresponding imaging text reports [44]. The labeling functions in this example are clinician-designed, expressing simple pattern-matching or ontology-lookup heuristics.

c) Non-Clinical Text Classification: The **Crowdsourcing**, **Recsys**, and **Spam** datasets are publicly-available at www.snorkel.org. The objective of **Crowdsourcing** is to label tweets pertaining to weather expressing either a positive or negative sentiment. **Recsys** aims to predict whether a user will read and like any given book or not. Finally, the **Spam** dataset aims to classify spam emails.

B. Implementation

In this section, we provide details on the implementation of our approach for curating adversarially ordered natural datasets. The implementation of probabilistic labeling via majority vote is straightforward. For probabilistic labeling via a generative model, we use a tool called Snorkel to train a generative model. Snorkel is the state-of-the-art tool for weak label combination and has been applied to several applications. We use the latest version at the time of writing, version 0.9.7.

For our confidence intervals, we use the confidence interval for a binomial proportion implementation, namely the "proportion_confint" function, from the statsmodels Python library, version 0.14.0. We specify the hyperparameters such that the Clopper-Pearson interval based on the Beta distribution with a 5% significance level is used. Specifically, hyperparameters "alpha" and "method" are set to 0.05 and "beta", respectively.

Generative models do not generalize to unseen samples, so we combine the train, validation, and test splits (without labels) as training data for the generative model. In order to evaluate the accuracy of the noisy datasets produced by our approach, we select the samples to be included in our datasets from the available labeled data (i.e., train for the alarm suppression examples and the validation and test splits combined for all other examples).

The code, including the approach implementation and script to generate our results, and the case study data are available at https://github.com/sfpugh/Naturally-Adversarial-Datasets.

C. Comparative Approaches

We previously discussed two primary challenges of probabilistic labeling via weak supervision techniques: (1) labeling function dependence, and (2) unreliability of confidence scores output by probabilistic labelers. In our evaluation, we will demonstrate why addressing these challenges is necessary to achieve sufficient performance. Hence we define three comparative approaches as follows. **PL Conf with all LFs** produces datasets by ordering samples by the probabilistic labeler's confidences ($\hat{g}(X)$) computed using *all* provided labeling functions (regardless of dependencies), and then selecting the top-p percent of samples per dataset (as done in our approach). **PL Conf with Indep. LFs** repeats the same steps as **PL Conf with all LFs** but uses a set of independent labeling functions selected by the same procedure from our approach. Lastly, **CI LB w/ all LFs** is an instance of our approach that skips the labeling function pruning step, that is, it is our approach using *all* provided labeling functions.

D. Evaluation of Natural Adversarial Ordering

Finally, we present the results of our approach applied to several case studies. Recall that the goal of the paper is to generate a sequence of statistically valid adversarially ordered natural datasets as per Def. 3. We validate the progressive increase of "adversarialness" in the datasets by analyzing the Spearman's rank correlation (Def. 2) of the adversarially ordered datasets.

The parameters of our approach are set as follows. Correlation between two labeling functions greater than 0.5 indicates that the labeling functions are dependent, i.e., correlation threshold $\delta = 0.5$. We allow for a 5% chance of the confidence intervals not containing the true confidence in our probabilistic

Study Type	Case Study	Probabilistic Labeler (PL)	PL Conf with all LFs	PL Conf with Indep. LFs	CI LB with all LFs	Our Approach
Medical	HR Low	Majority Vote	-0.719	-0.903	-0.863	-0.730
		Snorkel	–	–	-0.827	–
	HR High	Majority Vote	0.818	-0.976	0.857	-1.000
		Snorkel	0.964	0.927	0.988	-0.806
	RR Low	Majority Vote	-0.997	-0.879	-0.997	-0.891
		Snorkel	-0.997	–	-0.997	-0.806
	RR High	Majority Vote	-1.000	-0.988	-1.000	-0.952
		Snorkel	-1.000	-0.952	-1.000	–
	SpO_2 Low	Majority Vote	–	0.891	–	–
		Snorkel	-0.988	–	–	–
	Cross Modal	Majority Vote	-1.000	-1.000	-0.879	-0.782
		Snorkel	-1.000	-0.988	–	-0.806
Non-medical	Crowdsourcing	Majority Vote	-0.864	-0.988	-0.864	-0.976
		Snorkel	-0.976	-0.988	-0.864	-0.976
	Recsys	Majority Vote	-1.000	-1.000	–	–
		Snorkel	-0.988	-0.988	–	–
	Spam	Majority Vote	-0.985	-0.985	-0.767	-0.767
		Snorkel	-0.673	-0.673	-0.656	-0.656

TABLE II

SPEARMAN'S RANK CORRELATION COEFFICIENTS OF THE ADVERSARIALLY ORDERED NATURAL DATASETS WITH STATISTICALLY VALID ADVERSARIAL ORDERING. WE DESIRE DATASETS WITH STATISTICALLY SIGNIFICANT MONOTONICALLY DECREASING ACCURACY. HENCE WE REPORT COEFFICIENTS WHERE THE CORRESPONDING P-VALUE IS BELOW 5% AND COLOR MONOTONIC DECREASE IN BLACK AND MONOTONIC INCREASE IN RED.

labels, i.e., significance level $\alpha = 0.05$. Lastly, we let the, i.e., adversarial ordering significance threshold $\gamma = 0.05$. We report the Spearman's rank correlation coefficients for the accuracies of the datasets with statistically valid adversarial ordering produced by our approach and by the comparative approaches in Table II. The complete table of coefficients and corresponding p-values and plots of the dataset accuracies for all datasets are in Appendix A.

Main Takeaway: The results in Table II demonstrate that, unlike other approaches, our approach did not yield a statistically valid non-adversarially ordered dataset using real-world medical data. The negative correlation coefficients in Table II correspond to adversarially ordered datasets that have (generally) decreasing accuracy of weak labels. We recall that the central premise of this work is that statistically decreasing accuracy must be assured for any adversarially ordered dataset since decreasing accuracy across the ordered datasets is the property that will ultimately be tested as a measure of robustness. Simply put, if an approach yields a positive correlation coefficient in any application, it is unattractive as a method of curating adversarially ordered datasets. The results demonstrate that our approach yields statistically significant adversarially ordered datasets in twelve cases, and statistically invalid datasets on all other examples. We interpret this as a substantive outcome: our approach is applicable in a majority of the examples, and never yields datasets with statistically valid increasing accuracy. All the comparative approaches produce at least as many datasets with statistically valid decreasing accuracy, but also produce datasets with statistically significant monotonically increasing accuracy — thus violating our requirements. If used for robustness evaluation of learning models in practice, such datasets can cause incorrect and misleading results. Hence, we conclude that our approach yields more reliable results than the comparative approaches.

VI. DISCUSSION

In this discussion we review the impact and methods for improving the curation of adversarial ordered datasets and provide a discussion of the limitations of this technique.

A. Adversarial datasets need not be perfectly ordered.

Our approach aims to construct adversarially ordered natural datasets where, by construction, the label accuracy of each dataset progressively decreases. However, perfect ordering (i.e., a Spearman's Rank Correlation coefficient of exactly -1) is not required to evaluate robustness on our datasets; a statistically valid decreasing trend is sufficient (i.e., coefficient is negative and p-value is sufficiently low). When evaluating a learning-enabled component, we will observe the trend in its accuracy on these datasets and test that the trend is both decreasing and significant. This pragmatic approach aligns with the practical nature of real-world data, where inherent complexities and uncertainties may lead to variations in label accuracy. Thus, by focusing on the presence of a significant negative trend, we account for the genuine challenges posed by naturally occurring adversarial examples, making our evaluation more realistic and applicable to real-world scenarios.

B. Improving significance via data availability and weak labeling functions.

To comprehensively explore the real-world applicability of our method for curating adversarially ordered natural datasets, it is important to understand the contexts under which our method is ideally suited. Recall that our method constructs its datasets by sampling from the unlabeled input data. Consequently, the quantity and quality of the input data can have a significant impact on the method's efficacy. A larger input dataset, for example, typically provides a more diverse and representative sample of the underlying data distribution. As

a result, it can include a broader set of natural adversarial examples that our method can use. Additionally, recall that our method relies heavily on probabilistic labels produced by weakly-supervised data labeling techniques to identify the natural adversarial examples. Larger input data typically improves the accuracy of weakly-supervised data labeling techniques, and consequently the probabilistic labels. Since our method relies heavily on these labels, improving their accuracy can directly improve the quality of our datasets. However, as mentioned previously, data quality is also an important factor. Even when given more input data, such data should have limited bias and errors.

Labeling function quality is also an important factor for our method's applicability. Weak labels output by labeling functions are the foundation upon which weakly-supervised data labeling techniques learn to how to label data. Hence low-accuracy labeling functions can lead to inaccurate probabilistic labels, which is likely to cause our approach to construct improper datasets. A common assumption of weakly-supervised data labeling techniques are that the provided labeling functions are at least 50% accurate, but higher accuracy is generally more desirable. Unfortunately, how to write quality labeling functions is an open area research area [45], [46]. However, accurate labeling function design is not within the scope of this paper thus we assume the engineer has ensured the quality of the supplied labeling functions. Our approach also requires that there exists an independent subset of at least three labeling functions among those supplied. In practice, however, labeling functions often exhibit many statistical dependencies and can limit the applicability of our approach.

In summary, the curation of adversarial ordered datasets can be further improved by:

- Increasing the number of unlabeled examples
- Increasing the number and accuracy of labeling functions

C. Significance detection remains an open challenge.

As observed in Section V, our approach can yield adversarially ordered natural datasets with statistically insignificant ordering. Such datasets should not be used for robustness analysis as it is unclear what the expected trend of a learning-enabled component's accuracy should be on datasets with insignificant adversarial ordering. For the evaluation in this paper, we determined statistical significance of our datasets by leveraging the ground-truth labels of the input data. Ground-truth labels are important for determining the expected output, the basis of a comparison to the observed outcomes to quantify significance. Our high-level goal is to provide a method for evaluating robustness to natural adversarial examples when ground-truth labels are unavailable or prohibitively expensive to obtain. Unfortunately, the development of methods for testing statistical significance without ground truth labels is an open area of research. We plan to explore this in future work.

D. Potential Commercial Value.

This work presents a method for curating naturally adversarial ordered datasets for learning-enabled medical CPS. But, the fact remains that the gold standard for evaluating safety and efficacy of learning-enabled medical CPS is a clinical trial. Unfortunately, one of the most expense aspects of medical CPS development is real-world experimentation (i.e., observational/clinical trials) – often requiring years of engineering development and obtaining regulatory approvals to execute. Different from traditional labeled adversarial dataset robustness analysis, the approach presented herein is light weight – only requiring access to unlabeled (previously collected) examples and weak labeling functions. This is a commercial asset of the proposed approach to early stage development of learning-enabled medical CPS technologies (i.e., before labeled observational data collection). Leveraging the work herein, it is now feasible to evaluate the robustness in several commercially important scenarios. For example, an unlabeled dataset with different demographics than the training set could be used to assess whether a learning-enable medical CPS exhibits inherent demographic bias – a major issue in modern medical technology development. Another example is the proposed technique can be used in coordination with unsupervised techniques to assess real-world robustness in the absence of labeled data. Lastly, in the commercial development of learning-enabled medical CPS technologies – due to the pressures of raising capital and healthcare economics – it is important to "fail fast" (i.e., to quickly rule out ideas that are unlikely to succeed) prior to incurring significant development costs. This work provides a technique that may help early-stage researchers and entrepreneurs alike identify foundational robustness flaws in their approach prior to expensive data collection.

VII. Conclusion

In this paper, we proposed an approach for curating a sequence of adversarially ordered natural datasets for the purpose of evaluating model robustness to natural adversarial examples. Our approach identifies a set of independent labeling functions to use for probabilistic labeling. Probabilistic labels obtained via weak supervision techniques were used as proxy of the unknown true labels. We quantify the uncertainty in these labels with Clopper-Pearson confidence bounds, and construct our datasets according to the lower bound which is a indication of how "naturally adversarial" a sample may be. Finally, we evaluated our approach on six clinical case studies and three others and showed that we successfully produce natural datasets with statistically valid adversarial ordering, and do not produce datasets with statistically valid non-adversarial ordering. Directions for future work include (1) evaluating the robustness of real classifiers using our statistically valid adversarially ordered natural datasets, (2) devising a significance detector for adversarial ordering, (3) developing weakly-supervised methods for evaluating additional properties of deep learning models, and (4) calibrating weakly-supervised data labeling techniques.

Acknowledgment

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

REFERENCES

[1] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Computer methods and programs in biomedicine*, vol. 161, pp. 1–13, 2018.

[2] M. A. Morid, O. R. L. Sheng, and J. Dunbar, "Time series prediction using deep learning methods in healthcare," *ACM Trans. on Management Information Systems*, vol. 14, no. 1, pp. 1–29, 2023.

[3] I. Lee and O. Sokolsky, "Medical cyber physical systems," in *Proc. of the 47th design automation conference*, 2010, pp. 743–748.

[4] A. Watson, J. Park, S. Pugh, O. Sokolsky, J. Weimer, and I. Lee, "Medical cyber-physical systems: Iomt applications and challenges," in *Proc. of Asilomar*, 2022, pp. 998–1004.

[5] 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.

[6] B. J. Drew, P. Harris, J. K. Zègre-Hemsey, T. Mammone, D. Schindler, R. Salas-Boni, Y. Bai, A. Tinoco, Q. Ding, and X. Hu, "Insights into the problem of alarm fatigue with physiologic monitor devices: a comprehensive observational study of consecutive intensive care unit patients," *PloS one*, vol. 9, no. 10, p. e110274, 2014.

[7] P. Lameski, E. Zdravevski, S. Koceski, A. Kulakov, and V. Trajkovik, "Suppression of intensive care unit false alarms based on the arterial blood pressure signal," *IEEE Access*, vol. 5, pp. 5829–5836, 2017.

[8] H. Nguyen, S. Jang, R. Ivanov, C. Bonafide, J. Weimer, and I. Lee, "Reducing pulse oximetry false alarms without missing life-threatening events," *Smart Health*, vol. 9-10, 07 2018.

[9] W.-T. M. Au-Yeung, A. K. Sahani, E. M. Isselbacher, and A. A. Armoundas, "Reduction of false alarms in the intensive care unit using an optimized machine learning based approach," *NPJ digital medicine*, vol. 2, no. 1, pp. 1–5, 2019.

[10] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdić, "Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3810–3822, 2018.

[11] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao *et al.*, "Wilds: A benchmark of in-the-wild distribution shifts," in *Proc. of ICML*. PMLR, 2021, pp. 5637–5664.

[12] C. P. Bonafide, A. R. Localio, J. H. Holmes, V. M. Nadkarni, S. Stemler, M. MacMurchy, M. Zander, K. E. Roberts, R. Lin, and R. Keren, "Video analysis of factors associated with response time to physiologic monitor alarms in a children's hospital," *JAMIA Pediatrics*, vol. 171, no. 1, pp. 524–531, 2017.

[13] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proc. of CVPR*, 2021, pp. 15 262–15 271.

[14] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, and C. Ré, "Data programming: Creating large training sets, quickly," *NeurIPS*, vol. 29, 2016.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[16] Y. Vorobeychik and M. Kantarcioglu, *Adversarial Machine Learning*, R. Brachman, Ed. Cham, Switzerland: Morgan & Claypool Publishers, Aug. 2018.

[17] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," *arXiv preprint arXiv:1801.02612*, 2018.

[18] S. Chen, O. Sokolsky, J. Weimer, and I. Lee, "Data-driven Adaptive Safety Monitoring using Virtual Subjects in Medical Cyber-Physical Systems: A Glucose Control Case Study," *Journal of Computer Science and Engineering*, pp. 75–84, Sep. 2016.

[19] T. Kushner, B. Wayne Bequette, F. Cameron, G. Forlenza, D. Maahs, and S. Sankaranarayanan, "Models, devices, properties, and verification of artificial pancreas systems," *Automated Reasoning for Systems Biology and Medicine*, pp. 93–131, 2019.

[20] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," *arXiv preprint arXiv:1703.09387*, 2017.

[21] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," 2019.

[22] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," in *Proc. of DMBDA*. Springer, 2022, pp. 409–423.

[23] K.-K. Sung, "Learning and example selection for object and pattern detection," 1996.

[24] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," *Communications of the ACM*, vol. 64, no. 9, pp. 99–106, 2021.

[25] C. Bhagavatula, R. L. Bras, C. Malaviya, K. Sakaguchi, A. Holtzman, H. Rashkin, D. Downey, S. W.-t. Yih, and Y. Choi, "Abductive commonsense reasoning," *arXiv preprint arXiv:1908.05739*, 2019.

[26] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?" *arXiv preprint arXiv:1905.07830*, 2019.

[27] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner, "Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs," *arXiv preprint arXiv:1903.00161*, 2019.

[28] Y. Bisk, R. Zellers, J. Gao, Y. Choi *et al.*, "Piqa: Reasoning about physical commonsense in natural language," in *Proc. of AAAI*, vol. 34, no. 05, 2020, pp. 7432–7439.

[29] D. Hendrycks, C. Burns, S. Basart, A. Critch, J. Li, D. Song, and J. Steinhardt, "Aligning ai with shared human values," *arXiv preprint arXiv:2008.02275*, 2020.

[30] C. C. Aggarwal, *Outlier Analysis*. New York: Springer, Jan. 2013.

[31] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proc. of the IEEE*, vol. 109, no. 5, pp. 756–795, 2021.

[32] 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.

[33] A. Pal and V. N. Balasubramanian, "Adversarial data programming: Using gans to relax the bottleneck of curated labeled data," in *Proc. of CVPR*, 2018, pp. 1556–1565.

[34] ——, "Generative adversarial data programming," *arXiv preprint arXiv:2005.00364*, 2020.

[35] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, vol. 100, no. 3/4, pp. 441–471, 1987.

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

[37] S. H. Bach, B. He, A. Ratner, and C. Ré, "Learning the structure of generative models without labeled data," in *Proc. of ICML*. PMLR, 2017, pp. 273–282.

[38] 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 *Proc. of ICML*. PMLR, 2020, pp. 3280–3291.

[39] 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 *Proc. of CHASE*. IEEE, 2021, pp. 70–81.

[40] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. of ICML*. PMLR, 2017, pp. 1321–1330.

[41] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial," *Biometrika*, vol. 26, no. 4, pp. 404–413, 1934.

[42] M. MacMurchy, S. Stemler, M. Zander, and C. P. Bonafide, "Research: Acceptability, feasibility, and cost of using video to evaluate alarm fatigue," *Biomedical instrumentation & technology*, vol. 51, no. 1, pp. 25–33, 2017.

[43] S. Pugh, I. Ruchkin, C. Bonafide, S. Demauro, O. Sokolsky, I. Lee, and J. Weimer, "Evaluating alarm classifiers with high-confidence data programming," *ACM Trans. Comput. Healthcare*, vol. 3, no. 4, nov 2022.

[44] 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.

[45] P. Varma and C. Ré, "Snuba: Automating weak supervision to label training data," in *Proc. of VLDB*, vol. 12, no. 3, 2018, p. 223.

[46] N. Das, S. Chaba, R. Wu, S. Gandhi, D. H. Chau, and X. Chu, "Goggles: Automatic image labeling with affinity coding," in *Proc. of SIGMOD*, 2020, pp. 1717–1732.

Case Study	Probabilistic Labeler (PL)	PL Conf with all LFs	PL Conf with Indep. LFs	CI LB with all LFs	Our Approach
HR Low	Majority Vote	-0.719 (0.019)	-0.903 (0.000)	-0.863 (0.001)	-0.730 (0.017)
	Snorkel	-0.313 (0.379)	-0.730 (0.017)	-0.827 (0.003)	0.595 (0.070)
HR High	Majority Vote	0.818 (0.004)	-0.976 (0.000)	0.857 (0.002)	-1.000 (0.000)
	Snorkel	0.964 (0.000)	0.927 (0.000)	0.988 (0.000)	-0.806 (0.005)
RR Low	Majority Vote	-0.997 (0.000)	-0.879 (0.001)	-0.997 (0.000)	-0.891 (0.001)
	Snorkel	-0.997 (0.000)	-0.394 (0.260)	-0.997 (0.000)	-0.806 (0.005)
RR High	Majority Vote	-1.000 (0.000)	-0.988 (0.000)	-1.000 (0.000)	-0.952 (0.000)
	Snorkel	-1.000 (0.000)	-0.952 (0.000)	-1.000 (0.000)	-0.455 (0.187)
SpO_2 Low	Majority Vote	-0.309 (0.385)	0.891 (0.001)	-0.248 (0.489)	0.188 (0.603)
	Snorkel	-0.988 (0.000)	0.430 (0.214)	-0.455 (0.187)	-0.612 (0.060)
Cross Modal	Majority Vote	-1.000 (0.000)	-1.000 (0.000)	-0.879 (0.001)	-0.782 (0.008)
	Snorkel	-1.000 (0.000)	-0.988 (0.000)	-0.624 (0.054)	-0.806 (0.005)
Crowdsourcing	Majority Vote	-0.864 (0.001)	-0.988 (0.000)	-0.864 (0.001)	-0.976 (0.000)
	Snorkel	-0.976 (0.000)	-0.988 (0.000)	-0.864 (0.001)	-0.976 (0.000)
Recsys	Majority Vote	-1.000 (0.000)	-1.000 (0.000)	-0.321 (0.365)	-0.321 (0.365)
	Snorkel	-0.988 (0.000)	-0.988 (0.000)	-0.539 (0.108)	-0.539 (0.108)
Spam	Majority Vote	-0.985 (0.000)	-0.985 (0.000)	-0.767 (0.010)	-0.767 (0.010)
	Snorkel	-0.673 (0.033)	-0.673 (0.033)	-0.656 (0.039)	-0.656 (0.039)

TABLE III

SPEARMAN'S RANK CORRELATION OF THE ADVERSARIALLY ORDERED NATURAL DATASETS. THE VALUES REPORTED ARE THE CORRELATION COEFFICIENT AND CORRESPONDING P-VALUE IN PARENTHESES. WE DESIRE DATASETS WITH STATISTICALLY SIGNIFICANT MONOTONICALLY DECREASE. HENCE WE APPLY A 5% SIGNIFICANCE THRESHOLD TO THE P-VALUES, AND COLOR STATISTICALLY SIGNIFICANT AND NEGATIVE COEFFICIENTS IN GREEN AND COLOR STATISTICALLY SIGNIFICANT AND POSITIVE COEFFICIENTS IN RED.

APPENDIX

A. Additional Results

Table III shows the full results, i.e., the Spearman's Rank Correlation coefficients and p-values of the adversarially ordered natural datasets produced by our approach and comparative approaches across our nine case studies. Figure 3 shows the accuracy of the adversarially ordered natural datasets produced by our approach and comparative approaches across our nine case studies. For each case study, we curate ten datasets (i.e., $N = 10$) and then plot their accuracy scores. We also quantify the uncertainty in these accuracies by placing a 90% binomial confidence interval around them. For each dataset $D_i = (X_i, Y_i, \hat{Y}_i)$ where $i \in \{1, \ldots, N\}$, the width of the confidence interval is computed as,

$$1.64 \cdot \sqrt{\frac{\text{ACC}(Y_i, \hat{Y}_i) \cdot \left(1 - \text{ACC}(Y_i, \hat{Y}_i)\right)}{|Y_i|}}$$

where

$$\text{ACC}(Y, \hat{Y}) = \frac{1}{|Y|} \sum_{i=1}^{|Y|} \mathbf{1}\left(y_i = \hat{y}_i\right)$$

and $|\cdot|$ denotes the cardinality of the given set.

B. Repeatability

The implementation of our approach to curating adversarially ordered natural datasets is available on Github at https://github.com/sfpugh/Naturally-Adversarial-Datasets. The approach steps are implemented in `Naturally_Adversarial_Datasets/curate_datasets.py`. We provide the experimental data in `data.tar.gz` in the repeatability package materials. A script to reproduce the complete results (i.e., Table III) is provided in `scripts/generate_results.sh`. Note, the results in Table II show a subset of the values in Table III, thus the script produces this table as well.

The code can be run with Python 3.8 and packages `snorkel` (v0.9.7) and statsmodels (v0.14.0).

To create the Docker image, use the following commands:

```
git clone https://github.com/sfpugh/Naturally-Adve
cd Naturally_Adversarial_Datasets
docker build -t nad .
```

To run the Docker container and mount the data `data.tar.gz`, use the following commands:

```
docker run -v path/to/data.tar.gz:/app/Naturally_A
```

Prior to running the code, extract files from `data.tar.gz` in the `data` directory.

```
cd data
tar -xzvf data.tar.gz
cd ..
```

To reproduce the values in Table III (and Table II) run `generate_results.sh` in the `scripts` directory.

```
cd scripts
bash generate_results.sh
```

Fig. 3. Accuracy of datasets produced by our approach and comparative approaches. Recsys and Spam do not have dependent labeling functions hence the lines for PL with all LFs and PL with Indep. LFs, and CI LB with all LFs and our approach overlap.