Evaluating Alarm Classifiers with High-confidence Data Programming

SYDNEY PUGH and IVAN RUCHKIN, University of Pennsylvania
CHRISTOPHER BONAFIDE and SARA DEMAURO, Children's Hospital of Philadelphia
OLEG SOKOLSKY, INSUP LEE, and JAMES WEIMER, University of Pennsylvania

Classification of clinical alarms is at the heart of prioritization, suppression, integration, postponement, and other methods of mitigating alarm fatigue. Since these methods directly affect clinical care, alarm classifiers, such as intelligent suppression systems, need to be evaluated in terms of their sensitivity and specificity, which is typically calculated on a labeled dataset of alarms. Unfortunately, the collection and particularly labeling of such datasets requires substantial effort and time, thus deterring hospitals from investigating mitigations of alarm fatigue. This article develops a lightweight method for evaluating alarm classifiers without perfect alarm labels. The method relies on probabilistic labels obtained from data programming—a labeling paradigm based on combining noisy and cheap-to-obtain labeling heuristics. Based on these labels, the method produces confidence bounds for the sensitivity/specificity values from a hypothetical evaluation with manual labeling. Our experiments on five alarm datasets collected at Children's Hospital of Philadelphia show that the proposed method provides accurate bounds on the classifier's sensitivity/specificity, appropriately reflecting the uncertainty from noisy labeling and limited sample sizes.

CCS Concepts: • **Computing methodologies** → **Semi-supervised learning settings**; • **Applied computing** → **Health care information systems;**

Additional Key Words and Phrases: Physiological alarms, weak supervision, data programming, PAC bounds

ACM Reference format:
Sydney Pugh, Ivan Ruchkin, Christopher Bonafide, Sara DeMauro, Oleg Sokolsky, Insup Lee, and James Weimer. 2022. Evaluating Alarm Classifiers with High-confidence Data Programming. *ACM Trans. Comput. Healthcare* 3, 4, Article 43 (October 2022), 24 pages.
https://doi.org/10.1145/3549942

1 INTRODUCTION

Alarm fatigue is a well-known problem of physiologic monitoring in the hospital setting [11]. Bedside monitors continuously measure heart rhythm, heart rate, respiratory rate, blood oxygen, and other signals—and produce an alarm when the signals appear abnormal. Many of these alarms are not actionable or informative and often overwhelm the caregivers. The end result is that the clinicians react slowly, if at all, to alarms that have a small

This work was supported in part by NSF-1915398 and NIH R18 HS026620.
Authors' addresses: S. Pugh, I. Ruchkin, O. Sokolsky, I. Lee, and J. Weimer, University of Pennsylvania, Philadelphia, PA; emails: {sfpugh, iruchkin, sokolsky, lee, weimerj}@seas.upenn.edu; C. Bonafide and S. DeMauro, Children's Hospital of Philadelphia 3401 Civic Center Blvd, Philadelphia, PA 19104; emails: {bonafide, demauro}@chop.edu.

2637-8051/2022/10-ART43 $15.00
https://doi.org/10.1145/3549942

but nonzero probability of representing a critical patient need [5]. Ideally, the clinicians should only be alerted by the most important and actionable alarms, be able to interact with other informative alarms as they prefer, whereas the rest of the alarms are suppressed.

Alarm fatigue can be mitigated by reducing or prioritizing alarms via threshold tuning, customization, delaying, integration, and other methods [18]. Researchers have proposed novel algorithms for monitoring and suppressing unnecessary alarms based on advanced data processing [1, 8, 10]. At the core of algorithmic methods of reducing alarm fatigue is the *classification of alarms* into multiple classes such as priority, actionability, informativeness, suppressibility, delayability, and so on. This classification needs to be carefully balanced with the possibility of missing critically important and urgent alarms.

The clinical investigation and deployment of alarm classification systems is predicated on their expected performance. For example, when deploying an alarm suppression system, hospital policy makers need to ensure that its specificity to critical alarms is above a certain level, so that most of the critical alarms will continue to be reported. Measuring the performance of an alarm classifier typically requires a representative dataset of alarms correctly labeled with the classes of interest (e.g., suppressible vs. non-suppressible).

It is time-consuming and expensive to create highly accurate labeled datasets for the evaluation and tuning of alarm classifiers. A common way to do so is to perform an observational study [3] of many patients and manually label each period of time when a certain type of alarm would be, for instance, actionable. Such a study is a major commitment when it comes to an initial deployment of a novel alarm classifier, in part due to the significant effort of manually labeling thousands of alarms. Furthermore, it is impractical to perform an observational study for every adjustment of the settings of an alarm classifier throughout its lifecycle. This cost can be reduced with patient simulations [7], but precise and realistic simulations of human physiology are notoriously difficult and expensive to construct.

This article proposes a cheap and rapid method of estimating the performance of a clinical alarm classification system *in the absence of a dataset with highly accurate labels*. Our method can support early-stage low-cost evaluations of alarm classifiers in a variety of ways. For example, it can prioritize observational studies of classifiers with a higher potential to alleviate alarm fatigue so that the manual labeling effort is spent optimally. It can also guide the tuning of the classifier's settings towards effective alarm management, reducing the risk of missing critically important alarms.

A key step of our method is to probabilistically label patient data according to the emerging paradigm of *data programming* [16]. We start with a dataset of unlabeled patient data, typically abundant in most clinical settings, several alarm classes of interest, and an alarm classifier with tunable settings. We then collect clinical intuitions about this alarm type and encode them as *labeling functions*—weak classifiers of these alarms that can abstain and need not be comprehensive or non-contradictory. Often, a labeling function represents an intuitive heuristic such as "if the heart rate is high, then the alarm is actionable". The outputs of the labeling functions can be combined in a variety of ways, via majority vote or a *generative model*, resulting in labels of varied confidence for each data point. Finally, using the *high-confidence subset* of the labeled data, we estimate the uncertain *true class-wise rates* (e.g., sensitivity and specificity if we consider only two classes of alarms) of the alarm classifier.

To communicate the uncertainty of our estimates, we mathematically develop *confidence bounds* on the true rates of the alarm classifier. These bounds indicate, for a given level of confidence, the interval of possible values for the true rate that one could obtain from a hypothetical observational study of a given size. These bounds account for a finite number of samples for each class, potentially incorrect weak labels, and the uncertainty in estimating the confidence in our weak labels. Accounting for the confidence estimation uncertainty is a novel contribution in this journal extension of our earlier work [14].

We validated our method with five datasets of clinical alarms, derived from a 551-hour alarm labeling effort from Children's Hospital of Philadelphia. One of these datasets was used in our prior work [14], while the remaining four allow for novel experiments. We evaluated the performance of alarm actionability classifiers that threshold the key physiological signals such as SpO_2 and heart rate. Our method's estimated confidence

bounds almost always contain the true-label-based specificity and sensitivity—and vary appropriately based on the amount of available data. This application demonstrated our method of estimating the sensitivity-specificity tradeoffs in an alarm classifier without investing hundreds of hours into labeling the alarm data.

In summary, this article makes three research contributions:

— A data programming-based method for estimating the true rates of alarm classifiers.
— Confidence bounds on the true rate estimates, accounting for the uncertainty in sampling, labeling, and confidence estimation.
— A successful application of the above method to five clinical alarm datasets.

The remainder of this article proceeds as follows. The next section presents a motivating scenario for low-cost evaluation of alarm classifiers. Section 3 discusses the literature related to evaluating alarm classifiers. Section 4 introduces the minimal background and precisely formulates the mathematical problem at the heart of our method, which then is described in the following section. The five alarm datasets of are described in Section 6, and its results can be found in Section 7. The article concludes with a summary in Section 8.

2 MOTIVATING SCENARIO

This article focuses on clinical alarms produced by physiological monitors in a hospital setting. By an "alarm" we mean a combination of waveforms and patient data that have triggered a monitor to notify the clinicians. An *alarm classifier* is some algorithm or device that categorizes the alarms in a clinically useful manner. For example, one algorithm may determine which alarms are actionable and prioritize their delivery to the nursing staff. Another algorithm may find suppressible alarms and remove them from the feed. Yet another algorithm may discover equipment malfunction alarms (as opposed to the patient-related ones) and notify the technical personnel or the manufacturer. Our work considers all such algorithms to be alarms classifiers.

Most alarm classifiers have a direct impact on patient treatment and, therefore, need to be evaluated to justify their deployment and use. For example, to justify using an alarm suppression system, the decision-makers need to ensure that it suppresses some fraction of false alarms and avoids suppressing too many true alarms. This evaluation would quantify the *true class-wise rates*: the proportion of samples labeled correctly for each class. In a two-class setting, the true rates are *sensitivity* (for the positive class) and *specificity* (for the negative class).

Consider a scenario to motivate our work. A hospital is considering the deployment of an alarm suppression system, which has tunable parameters. The effect of these parameters on the hospital's population of patients is not known: will the reduction in superfluous alarms be worth the risk of missing important and actionable alarms? To answer this question, the hospital would need to perform a controlled observation study in which a sample of alarms is collected, labeled, and used to test the suppression system with different parameter settings. Such a study is particularly time-consuming because the collected alarms (likely thousands of them) would need to be manually labeled. This study may indicate that the expected suppression performance is far below acceptable, and the labeling effort would be wasted. On the other hand, the classifier may perform so well that the hospital may decide to skip the observational study and, instead, advance to an interventional pilot study.

Thus, the hospital could benefit from cheaper and faster ways of predicting the range of *hypothetical outcomes of an observation study* for an alarm suppression system, ideally avoiding the manual labeling of alarms. As the hospital continues to collect alarm data, it wants to gain increasing confidence in its prediction, until the point when some decision can made. This scenario is possible not only for alarm suppression systems, but also for other algorithmic *classifiers of alarms*, for example, whether an alarm is actionable, what priority it should have, or whether it was caused by clinical events or technical issues. Since the classes of alarms and patient populations differ and change over time, clinicians need to repeatedly predict classifier performance, and therefore a general estimation approach is needed. This general scenario of evaluating an alarm classifier without perfect labels is shown in Figure 1.

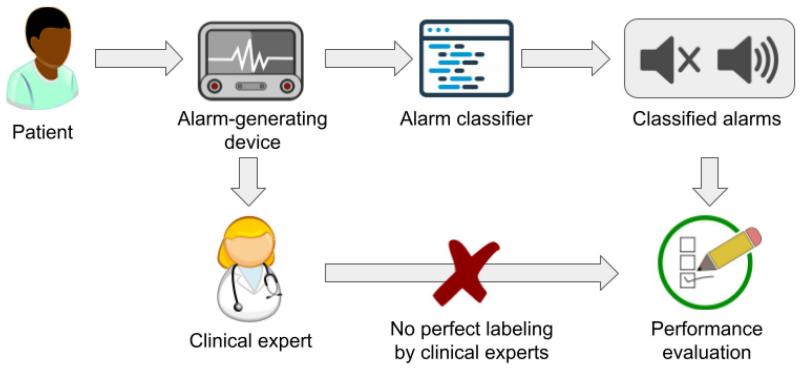

Fig. 1. A motivating scenario: An alarm classifier needs to be evaluated without true labels of alarms.

3 RELATED WORK

There exists a vast literature on clinical alarms and unsupervised/weakly-supervised learning. We focus on the areas particularly relevant to our goal of evaluating alarm classifiers.

Alarm fatigue is a serious and well-known problem of physiological monitors [5, 11]. To address it, a variety of approaches to suppress or prioritize alarms have been proposed based on techniques from signal processing, statistics, and machine learning [1, 8, 10]. Many of such approaches need patient data labels to be designed or trained, and all of them need the labels to be evaluated. Our article introduces a lightweight way of performing these evaluations without investing in high-quality labels. Note that our proposed method is not specific to any physiological input, unlike many alarm classification techniques.

The gold standard for evaluating the clinical effectiveness of an alarm classifier is an interventional study, in which the researchers deploy the system and measure its effects compared to a control group. To estimate the true rates of a classifier (e.g., its sensitivity and specificity), a controlled observational study would be sufficient: alarms are collected from physiological monitors, manually labeled by the clinical experts, and fed into the classifier. Then the true rates are computed by comparing its outputs with the manual labels. Both types of studies require substantial time and effort, in part due to the need to label each alarm by hand. For example, nurses may need to review video feeds of patients to determine the actionability of alarms [9]. Our work is not meant to replace either type of studies; instead, we aim at prioritizing, guide, and reduce the risk of observational studies by providing an early and cheap estimation of the expected performance of an alarm classifier.

The **Area under the Receiver Operating Characteristic Curve** (**AuROC**) is a metric often used to evaluate the quality of medical classifiers. There exists literature where the AuROC can be estimated in the form of confidence intervals from parameters such as error rates and the number of positive/negative samples [4]. Another AuROC estimation method from test scores via bootstrapping [2]. However, these methods require a labeled dataset which is prohibitively expensive to collect. Our approach can be used to generate confidence intervals for sensitivity/specificity using a weakly labeled dataset, from which AuROC can be calculated.

Alarms can be labeled with high-precision methods using patient simulations and computer-aided clinical trials [6]. To provide realistic data, these methods require detailed physiological models, a building which is an expensive and lengthy task. For clinical alarms, an appropriate model is rarely available. Our method is related to observational studies in the same way as computer-aided clinical trials are related to traditional clinical trials. That is, we perform a virtual algorithmic evaluation of alarm classifiers. After that, our results can provide the basis for an observational study or a clinical trial of the algorithm or a device that classifies alarms.

Recently, a quick and inexpensive way of labeling data has emerged, known as *data programming* [16]. A key element of data programming 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, data programming algorithms produce probabilistic labels—a label and a confidence between 0 and 1—for each sample in the dataset. A prominent data programming tool *Snorkel* [15] estimates an optimal weight for each labeling function by using a generative graphical model and a prior on the class balance. Our approach encodes clinical intuitions about classes of alarms (e.g., suppressible vs. non-suppressible) as labeling functions, feeds them along with alarm data into Snorkel, and relies on the resulting probabilistic labels to quantify the uncertainty in the classifier performance.

Extending the data programming work, adversarial data programming generates data in addition to labeling it [12, 13]. A **Generative Adversarial Network** (**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. This technique is related to patient simulations and computer-aided clinical trials in that realistic labeled data can be generated for the purpose of evaluating an alarm classifier. Data generation can be used in an extension of our work to address applications with small datasets or imbalanced datasets.

This article builds on our previous approach to estimating true rates in alarm suppression systems [14]. Compared to that, we have extended the scope from evaluating suppression systems to evaluating any alarm classifier—and correspondingly extended our evaluation with four new alarm datasets of varying sizes. We have also extended the technical approach to account for the uncertainty in estimating the label confidence in a finite sample. This extension allows us to vary the bound width appropriately datasets with few samples, which did not satisfy the assumptions of the earlier approach.

4 PROBLEM FORMULATION

We start with an unlabeled dataset of alarms ($\mathcal{I}$) and consider a hypothetical, unavailable dataset of alarms ($\mathcal{I}^*$) of a known size collected during a hypothetical observation study from the same population as $\mathcal{I}$. We are evaluating an alarm classifier (S) that takes a sample and assigns one of J classes to it. This alarm classifier has tunable parameters affecting its output (e.g., an alarm threshold for high heart rate), and in the rest of the article, we treat each alarm classifier with different parameter values as a separate classifier S.

Our goal is to quantify how well the alarm classifier S would perform on the hypothetical dataset $\mathcal{I}^*$. We measure the classifier performance with its *true rate* for each class. For any class j, a true rate R_j is the fraction of samples labeled j that are indeed of class j. In the case of only two classes, the positive true rate is called *sensitivity*, and the negative true rate is called *specificity*. We aim at predicting the true rates for each class on the hypothetical dataset $\mathcal{I}^*$. We refer to these hypothetical, unknown rates as R_j^*, and each of them is, conceptually, determined on a subset of $\mathcal{I}^*$ containing all samples with label j, which we refer to as $\mathcal{I}_j^*$. Since we do not have access to $\mathcal{I}^*$, we will predict R_j^* indirectly using $\mathcal{I}$.

To use dataset $\mathcal{I}$, we first need to overcome the absence of true labels for it. So one sub-task will be to create a *probabilistic labeler*—an algorithm that takes a sample and assigns it an estimated label and a confidence in that label (a number between 0 and 1). We refer to the assigned label as $f(x)$ and its confidence as $g(x)$ for any sample x. We will build that labeler from *labeling functions*, which encode rules of thumb and heuristics acquired from clinicians. Each labeling function takes a sample and either abstains or assigns one of the classes to it. Labeling functions can contradict each other or abstain in different combinations. Once a probabilistic labeler is created, each sample x in the dataset $\mathcal{I}$ will be characterized by three values: a true (unknown) label $f^*(x)$, an estimated label $f(x)$, and a confidence $g(x)$ in the estimated label. In practice, we only have access to the latter two.

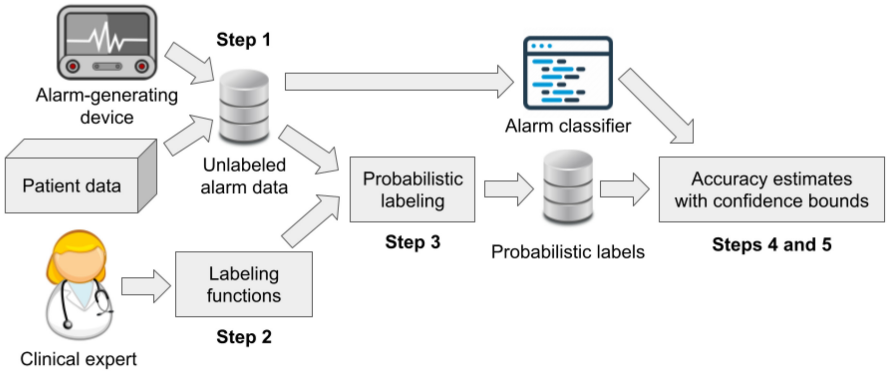

Fig. 2. Our approach of estimating the true rates of alarm classifiers.

Our second sub-task will be to structure $\mathcal{I}$ based on the estimated labels. For each label j, we form a *high-confidence set* $\mathcal{I}_j$ of samples x from $\mathcal{I}$ that the probabilistic labeler labeled with j (i.e., $f(x) = j$) and high confidence, (i.e., $g(x) \geq 1 - \epsilon$ for some constant ϵ that we pick up front).

On each high-confidence set $\mathcal{I}_j$, we can calculate the estimated true rate of S on class j, denoted as R_j. To predict the hypothetical rate R_j^*, we will come up with an *interval* C_j of possible true rates around R_j. This interval should contain the value of R_j^* with at least the *containment probability* p_j (a.k.a. the significance level), given to us by the user.

To summarize, the technical problem addressed in this article is to create the smallest interval containing the hypothetical rate R_j^* with a probability at least p_j. This problem can be stated mathematically as follows:

$$\min_{C_j} |C_j| \text{ subject to } \mathbb{P}(R_j^* \in C_j) \geq p_j.$$

5 TRUE RATE ESTIMATION APPROACH

This section describes our approach for producing confidence bounds for the class-wise true rates of an alarm classifier. Figure 2 summarizes the five steps of our approach:

(1) Collect unlabeled alarms and patient data
(2) Elicit heuristic labeling functions from clinicians
(3) Produce probabilistic labels for the collected alarms
(4) Estimate the average true rates of the alarm classifier
(5) Quantify confidence bounds around those rate estimates

5.1 Collecting Unlabeled Alarms

Our initial step is to collect a dataset of representative alarms and the corresponding patient data, which we will use to evaluate the alarm classifier. The patient data includes the static data (demographics, disease history, and so on.) and the vital signals that contextualize a raised alarm. Thus, there are two key aspects of data: determining which alarm instances to use and collecting the relevant patient and vitals data. More formally, we produce a set of representative unlabeled alarms data $\mathcal{I}$, with each datapoint corresponding to the features of an alarm that was raised for some patient.

When choosing the alarms, our goal is to get a sample from the representative distribution in a particular clinical setting. To this end, we use the state-of-the-art sampling approaches. Typically, alarms would be sampled from the sub-types targeted by the labeling system (e.g., technical or clinical, discussed in Section 6), based on their frequency during different times of the day, and appropriately from the target patient demographics.

The collection of patient and alarm data should be fully automated and, therefore, cheap. Given full automation, we aim for the data to be as comprehensive and unbiased as possible in a given clinical context. Richer datasets, such as those that include more patient information and diverse vitals, allow for more detailed labeling functions in the next step, ultimately improving the outcomes of our approach.

5.2 Eliciting Labeling Functions

We ask clinical experts (e.g., physicians and nurses) working in the relevant context to describe the guidelines that they use to make decisions on how an alarm should be classified (e.g., whether it is actionable or not). In the context of our case studies, we seek *quantifiable* guidelines for determining alarm suppressibility (e.g., if the heart rate is above 200 then actionable), as opposed to the difficult-to-quantify observations (e.g., if a child is kicking or moving then the alarm is not actionable) which are often used by clinicians. Such observations are difficult to encode in our data-driven approach and, hence, are excluded from our approach for now and left for future work.

The obtained guidelines are not perfect; that is, they output noisy labels, and our approach will account for that. Nonetheless, extremely inaccurate labeling functions can negatively affect the performance down the line. To address this, we ask clinicians stick to guidelines that, in their expert opinion, are *better than random chance* at identifying the alarm class correctly. Better-than-random labeling functions is a common requirement in data programming [16].

Each guideline is implemented as one or more labeling functions. Each labeling function takes patient data as input and either emits a label (suppressible/non-suppressible) or abstains for each sample in the unlabeled alarm dataset. Formally, the labeling functions Λ are elicited and applied to the data $\mathcal{I}$ to obtain the weak labels. For each data point, its weak label is a set of confidence for each of the J classes, and the highest-confidence class is understood to be the class label. We will refer to the weak labels of our data $\mathcal{I}$ as $L_\Lambda(\mathcal{I})$.

5.3 Probabilistic Labeling

In this step, we combine the weak labels from the labeling functions into a single "strong" probabilistic label. This label is characterized by a confidence value between 0 and 1, indicating the level of certainty regarding the label's accuracy. Mathematically, for each alarm x, we combine its weak labels $L_\Lambda(x)$ into a probabilistic strong label $f(x)$ with confidence $g(x)$.

Our framework computes the strong label and the confidence using a *weighted combination* over the weak labels with a fixed vector of weights w, which contains one weight per labeling function. Our framework allows for different techniques for weighted combination, and in this article, we use three of them:

- Majority vote
- Generative model with an uninformed prior
- Generative model with an informed prior

These three techniques are applied in two steps:

(1) Determine the weights w
(2) Combine weak labels using weights w

The goal of the first step is to learn a non-negative weight vector where each weight indicates the relative priority of the corresponding labeling function. The 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 the majority vote is uniform (i.e., $w = 1$ for all functions). The other two methods utilize a generative graphical model. Generative models are popular in state-of-the-art data programming literature [16]. Such models give higher weights to labeling functions with higher accuracies. Since the accuracies are unknown *a priori*, this model leverages the agreements and disagreements of the labeling functions in the data ($L_\Lambda(\mathcal{I})$) during its

training phase to estimate their accuracies. It can be trained with an uninformed prior (equal probability of every class) or with an informed prior (clinicians indicate the frequencies of each class of alarms). The technical details on this use of generative models can be found in our prior work [14].

In the second step, the learned weights are used to combine the weak labels into a probabilistic label for every alarm. Suppose now we want to label an alarm x given its weak labels $L_\Lambda(x)$ and the weights w. For the majority vote, since the weights are uniform, the confidence in class j $g(x)$ of a particular label is computed as the fraction of non-abstaining labeling functions choosing j. For generative models, we obtain class weights by adding up the weights of all labeling functions choosing that class. Then we pass the class weights through the softmax function, which is a standard way to normalize positive real numbers into a probability distribution. Finally, the highest normalized weight is used as the confidence $g(x)$.

The majority vote is a simple approach that is straightforward to apply. However, if many of the labeling functions are inaccurate, the label prediction can be incorrect with high confidence. The benefit of using generative models is that they can account for the inaccuracies of labeling functions and, thus, give a more accurate confidence.

5.4 Estimating Average True Rates

Now we estimate the average true rates of the alarm classifier using the probabilistic labels. That is, we are interested in finding a vector of numbers to estimate each of the J true rates given the probabilistic labels. Formally, for each j from 1 to J, we compute point estimates of R_j for an alarm classifier S given the data $\mathcal{I}$, weak labeler f, and confidence estimator g. The challenge here is to balance three uncertainties: the sampling uncertainty due to the limited size of $\mathcal{I}$, the weak labeling uncertainty in f from potentially inaccurate labeling functions, and confidence uncertainty that stems from a limited dataset where we estimate confidence g.

First, we need to pick the data that was labeled in a trustworthy manner. In our experience, using the whole dataset $\mathcal{I}$ is inadvisable because low-accuracy/low-confidence labels would significantly bias the outcome. Therefore, intuitively, we place more trust in samples that we label with high confidence (i.e., with low uncertainty about their true label). The threshold to qualify as a high confidence sample is different for each class to handle class imbalanced datasets. For each class j, we tolerate at most ϵ_j labeling uncertainty and, thus, only select samples with confidence at least $1 - \epsilon_j$. This leads us to construct *high-confidence subsets* $\mathcal{I}_j$ parameterized by ϵ_j, as defined earlier in Section 4.

The tradeoff between the sampling uncertainty and the labeling uncertainty is moderated by ϵ_j. For example, if we pick an arbitrarily small value of ϵ_j, our high-confidence set $\mathcal{I}_j$ will have few samples and large sampling uncertainty. On the other hand, if we pick a relatively large ϵ_j, the average confidence in $\mathcal{I}_j$ may be high, indicating a high labeling uncertainty. We will resolve this tradeoff in the next subsection by minimizing the combination of these uncertainties to get the tightest confidence bounds around our estimates.

Now, we are ready to estimate the average true rate on each class j by calculating the proportion of samples that our classifier labels j in the high-confidence set $\mathcal{I}_j$:

$$R_j = \frac{\sum_{x \in \mathcal{I}_j} \mathbb{1}\left(S(x) = j\right)}{|\mathcal{I}_j|}. \tag{1}$$

In a two-class setting, this formula gives us a point estimate of sensitivity ($j = 1$) and specificity ($j = 0$) of the alarm classifier.

5.5 Producing Confidence Bounds

Our average rate estimates from the previous subsection rely on finite samples and noisy labeling functions and, as a result, can be inaccurate. We quantify their potential inaccuracy by providing *confidence bounds* around our estimates—the intervals where the hypothetical true rate (i.e., one estimated from perfect labels in an observational study) could be found with some given confidence. More precisely, we interpret our confidence bound as

follows: it is an interval of likely true estimates using the correct labels of that class from a hypothetical observation study. That is, for class j, we aim at creating an interval $C_j = [R_j - c_j, R_j + c_j]$ containing, with probability of at least p_j, the hypothetical true rate R_j^* estimated from the manually (i.e., perfectly) labeled samples in $\mathcal{I}_j^*$. Notice that p_j can differ between the classes and, hence, reflect the acceptable level of clinical risk.

The interval size c_j needs to account for three uncertainties:

(1) *Sampling uncertainty:* the potential difference between random samples in $\mathcal{I}_j$ and $\mathcal{I}_j^*$.
(2) *Labeling uncertainty:* the potential inaccuracy of probabilistic labels of $\mathcal{I}_j$.
(3) *Confidence uncertainty:* the inaccuracy of the estimated average confidence in $\mathcal{I}_j$.

The first uncertainty will be estimated as a function of the sizes of $\mathcal{I}_j$ and $\mathcal{I}_j^*$ using the standard statistical bounds. For the second uncertainty, in our experience, with appropriate labeling functions discussed Section 5.2, high-confidence labels correspond to the low-uncertainty situations in which alarm classification is relatively consistent. We formalize this intuition with the following assumption over the underlying random variable $\tilde{x}$ that generates our alarms:

Assumption 1 (Consistent Rates across Datasets). *The true rates measured on high-confidence sets $\mathcal{I}_j$ do not differ in expectation from those measured on the manually-labeled sets $\mathcal{I}_j^*$ by more than the expected average uncertainty of the labels in $\mathcal{I}_j$:*

$$\left|\mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j^*\right] - \mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j\right]\right| \leq 1 - \mathbb{E}\left[g(\tilde{x}) \mid \tilde{x} \in \mathcal{I}_j\right].$$

Intuitively, our assumption means that, if the samples were labeled with confidence 95%, then the underlying average true rate on perfectly labeled samples could not differ by more than 5%.

For the third uncertainty, we derive a bound on the difference between the actual unknown average confidence $\mathbb{E}[g(\tilde{x}) \mid \tilde{x} \in \mathcal{I}_j]$ and our sample-based *estimate of average confidence* in dataset $|\mathcal{I}_j|$, denoted as $\eta_j = \frac{1}{|\mathcal{I}_j|}\sum_{n\in\mathcal{I}_j} g(x_n)$, using standard statistical bounds.

Putting together the three uncertainties, we derive the desired bound c_j on the difference between the estimates based on our probabilistic labels and the hypothetical-observation-study labels. The probability of exceeding that bound is given in the following theorem.

Theorem 1 (Bounded Difference of Accuracy Estimates). *For any class j, the difference between the probabilistic and manual estimates of true rate on class j exceeds bound c_j with a bounded probability:*

$$\mathbb{P}\left[\left|R_j(\mathcal{I}_j) - R_j(\mathcal{I}_j^*)\right| \geq c_j\right] \leq 6\exp\left(\frac{-2|\mathcal{I}_j||\mathcal{I}_j^*|}{\left(\sqrt{|\mathcal{I}_j|} + 2\sqrt{|\mathcal{I}_j^*|}\right)^2}(c_j + \eta_j - 1)^2\right),$$

where $\eta_j = \frac{1}{|\mathcal{I}_j|}\sum_{n\in\mathcal{I}_j} g(x_n)$ is the average label confidence in $\mathcal{I}_j$.

The proof can be found in Appendix A. This result means that the chance of our estimates disagreeing with the hypothetical estimates by more than c_j decreases with the increasing number of samples in $\mathcal{I}_j$ and $\mathcal{I}_j^*$, larger interval size c_j, and the higher confidence of our estimates η_j. This bound is contingent on the satisfaction of Assumption 1 about probabilistic labeling.

We want to guarantee that the hypothetical labeled estimate R_j^* is within c_j of our estimate R_j with probability p_j. Then, by equating p_j with the right-hand side of the inequality in Theorem 1 and expressing c_j in terms of p_j, we obtain the desired interval size c_j.

Corollary 1. *For any desired confidence p_j, the interval width c_j can be chosen as,*

$$1 - \eta_j + \sqrt{\frac{\left(\sqrt{|\mathcal{I}_j|} + 2\sqrt{|\mathcal{I}_j^*|}\right)^2}{-2|\mathcal{I}_j||\mathcal{I}_j^*|} \ln\left(\frac{p_j}{6}\right)}.$$

Finally, we return to the problem formulation from the end of Section 4: our goal is to minimize the size of the interval c_j given a fixed confidence p_j. So we optimize for the smallest interval over the choice of samples in $\mathcal{I}_j$ (by changing ϵ_j, which defines $\mathcal{I}_j$, which in turn determines η_j). Thus, we pick the interval size c_j as follows:

$$\min_{\mathcal{I}_j \subseteq \mathcal{I}} 1 - \eta_j + \sqrt{\frac{\left(\sqrt{|\mathcal{I}_j|} + 2\sqrt{|\mathcal{I}_j^*|}\right)^2}{-2|\mathcal{I}_j||\mathcal{I}_j^*|} \ln\left(\frac{p_j}{6}\right)}.$$

In summary, the presented results give us a way to produce uncertainty bounds for the true rates of the alarm classifier by putting a confidence bound around the accuracy estimates from Section 5.4. We pick the tightest interval given Theorem 1 while accounting for the sampling uncertainty, labeling uncertainty, and confidence estimation uncertainty.

6 CASE STUDIES

To evaluate the performance of our approach, we conducted experiments on five alarm datasets: low SpO_2, low/high **respiratory rate** (**RR**), and low/high **heart rate** (**HR**)[1] alarms. We consider alarm classifiers that label an alarm as either suppressible or non-suppressible. For *low* vital sign alarms, the classifier labels an alarm as suppressible if the vital sign measurement is *above* a specified threshold at the time of the alarm; otherwise it is labeled non-suppressible. For *high* vital sign alarms, the classifier labels an alarm as suppressible if the vital sign measurement is *below* a specified threshold, otherwise, it labels non-suppressible. We consider the context where the threshold of these alarm classifiers were improperly configured, resulting in a large volume of suppressible alarms being missed. Our goal is to establish and visualize the connection between the alarm classifiers' threshold and its sensitivity/specificity, given an unlabeled dataset of patient vitals data. This visualization will be compared with the sensitivity/specificity computed from the manually-annotated alarm data, which is our case stands in for a hypothetical observational study. In this section, we overview the dataset and data preprocessing approach, introduce the labeling functions collected for labeling alarms, and describe the implementation details of our approach.

6.1 Data

We used a de-identified dataset originally collected as part of a study approved by the Institutional Review Board of the Children's Hospital of Philadelphia (IRB #14-010846). Researchers video-recorded 551 hours of patient care on a medical unit at Children's Hospital of Philadelphia during July 2014 to November 2015 from 100 children whose families and nurses consented. In addition, the following data were collected: patient background information, all physiologic monitoring alarms with corresponding timestamps, and continuously recorded vital signs:

- Blood oxygen saturation (SpO_2) measured by a pulse oximeter,
- Pulse rate measured by a pulse oximeter,
- Heart rate measured by a 3-lead **electrocardiography** (**ECG**),
- Cardiac rhythm measured by a 3-lead ECG,
- Respiratory rate measured by a 3-lead ECG,
- **Noninvasive blood pressure** (**NBP**) measured by a cuff, from the physiologic monitoring network.

[1]These heart rate alarms were generated based on the pulse oximeter readings.

After the study, the researchers reviewed the alarms along with the video recordings and then annotated them with along three dimensions: technical versus clinical alarms, valid versus invalid alarms, and actionable versus non-actionable alarms [9]. Technical alarms indicate an issue with a physiologic monitor or its sensors, whereas clinical alarms indicate an issue with a patient's physiologic status (e.g., heart rate is too high). Valid alarms are those that correctly identify the physiologic status of a patient. Conversely, alarms that are false are considered invalid. A valid clinical alarm that results in or warrants clinical intervention or consultation can be further classified as actionable, otherwise non-actionable. Hence the alarms have the following annotations: technical alarms, invalid clinical alarms, valid actionable clinical alarms, and valid non-actionable clinical alarms.

A total of 9,547 clinical alarms of 26 different types are in the dataset. Low SpO_2, low/high respiratory rate, and low/high heart rate alarms account for 58% of total alarms, of which, 54% are invalid. Hence, calibrating a suppression classifier for these alarm types can help reduce alarm fatigue, and we focus on these alarms in our case study.

6.2 Data Preprocessing

Our analysis only considers a subset of the features of the original dataset:

- Patient age group: less than one month old, from one month to less than two months, from two months to less than six months, and six months and older.
- Patient vital signs: blood oxygen saturation, respiratory rate, heart rate measured by an ECG, and heart rate measured by a pulse oximeter—all measured at a maximum sampling rate of 0.2 Hz.
- Annotated low SpO_2, low/high respiratory rate, and low/high heart rate alarms with corresponding timestamps and durations.

As mentioned earlier, each alarm is annotated in terms of technical/clinical, valid/invalid, and actionable/non-actionable alarms. We interpret these labels with respect to suppressibility as follows. Technical alarms, valid non-actionable clinical alarms, and invalid alarms are interpreted as suppressible, whereas only valid actionable clinical alarms are non-suppressible.

6.3 Labeling Functions

We conducted two one-hour unstructured interviews with two pediatric physicians where we collected eighteen guidelines for deciding whether an alarm is suppressible or non-suppressible. Six of the guidelines are excluded from this study because the dataset does not have sufficient information to implement them. The guidelines are as follows:

(1) *Long alarm*: If the alarm duration is longer than t seconds, then the alarm is likely non-suppressible.
(2) *SpO_2 below the threshold for duration*: If SpO_2 is below threshold x for longer than t seconds since the alarm sounded, then the alarm is likely non-suppressible.
(3) *Heart rate above the threshold for duration*: If the heart rate is above threshold x for longer than t seconds since the alarm sounded, then the alarm is likely non-suppressible.
(4) *Heart rate below the threshold for duration*: If the heart rate is below threshold x for longer than t seconds since the alarm sounded, then the alarm is likely non-suppressible.
(5) *Respiratory rate below the threshold for duration*: If the respiratory rate is below threshold x for longer than t seconds since the alarm sounded, then the alarm is likely non-suppressible.
(6) *Repeat alarms*: If more than n alarms occurred within t seconds of the alarm, then the alarm is likely non-suppressible.
(7) *Short alarm*: If the alarm duration is less than t seconds, then the alarm is likely suppressible.

Table 1. Summary of the Five Alarm Datasets

Case Study	Number of Alarms	Percent Suppressible	Number of Datapoints
Low SpO_2	3,265	0.81	11,300
Low RR	312	0.35	527
High RR	574	0.25	968
Low HR	79	0.38	145
High HR	1,315	0.03	6,081

(8) *Immediate recovery*: If SpO_2 recovers to x within t seconds after the alarm sounds, then the alarm is likely suppressible.
(9) *Heart rate technical error*: If the difference between ECG heart rate and pulse oximeter heart rate is greater than x at the time of the alarm, then the alarm is likely suppressible.
(10) *Bad SpO_2 waveform*: If the SpO_2 waveform contains anomalies,[2] then the alarm is likely suppressible.
(11) *Bad heart rate waveform*: If the ECG heart rate waveform contains anomalies, then the alarm is likely suppressible.
(12) *Bad respiratory rate waveform*: If the respiratory rate waveform contains anomalies, then the alarm is likely suppressible.

From these guidelines, we instantiated *sixty-two labeling functions* for different values of parameters x, t, and n, picked in consultation with the two aforementioned physicians (see Appendix B). Of these labeling functions, 40 produce only suppressible labels, and the rest produce only non-suppressible labels.

6.4 Implementation

We implement the labeling functions as Python functions, taking in an alarm from the dataset and returning either a suppressible/non-suppressible label or an abstain. To generate probabilistic labels for the alarms in the dataset we use a tool called Snorkel to train our generative model [15]. 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 (www.snorkel.org).

The only hyperparameter we specify within Snorkel is the prior probability of labels. For the informed prior we use the exact proportion of suppressible/non-suppressible labels which were interpreted from the manual annotations; the priors are listed in the second column of Table 1. In cases when the prior is unknown, it can also be estimated from the labeling functions [17].

Our alarm samples for estimating sensitivity/specificity are generated from the vital sign readings at the moments when an alarm was sounding. For each case study, we consider the timestamps for which there is a measurement of the relevant vital sign (their counts are in the third column of Table 1). Then we map the labels assigned to the known alarms onto these samples. For each alarm, we assign all timestamps that occur during this alarm with its own labels (true and probabilistic). Lastly, we simulate applying the alarm classifier to the samples for different thresholds, and save the result as a label. Those thresholds are 0 to 100 for low SpO_2, 0 to 300 for low/high respiratory rate, and 0 to 200 for low/high heart rate. Thus for each sample we have,

— A timestamp,
— A vital sign measurement,
— A ground-truth label (from the original annotations),

[2]Waveforms with artifacts are generally unreliable. We look for anomalies (e.g., spikes and outliers) in the waveform to determine if it is bad or not.

— A label and its confidence from the generative model,
— A label from the alarm classifier for each threshold.

7 RESULTS

This section presents the results of our experiments described in the previous section.[3] Specifically, we evaluate the performance of the confidence bounds for the sensitivity and specificity of alarm suppression classifiers produced by our approach. We also evaluate the bounds for the tradeoff curve between sensitivity and specificity, because this curve may inform clinical choices of alarm classification parameters. A successful application of our approach would result in bounds that contain the hypothetical sensitivity/specificity and tighten when more data is made available. We compare our confidence bounds to those produced by the confidence uncertainty-unaware approach from earlier work [14].

The parameters of our approach are set as follows. We allow for a 10% chance of the confidence bounds not containing the true sensitivity/specificity, i.e., significance level $p_0 = p_1 = 0.1$. Label uncertainty ϵ_j determines the high-confidence subsets $\mathcal{I}_j$ that we optimize over in our approach when computing the confidence bounds. We search the space $\epsilon_j \in [0.001, 0.5]$ to find the tightest bounds.[4]

The confidence bounds for sensitivity, specificity, and the tradeoff between the two using different probabilistic labeling methods are depicted in Figures 3–5, respectively. To draw the tradeoff bounds, for each threshold we plot the $(specificity + c_0, sensitivity + c_1)$ for the upper-bound and $(specificity - c_0, sensitivity - c_1)$ for the lower-bound. Since we have access to true labels (i.e., the labels extracted from the alarm annotations), we use them to plot the *true curve* for the sensitivity/specificity/tradeoff, which represents the results of an observational study. Furthermore, we use the true labels to implement a fourth probabilistic labeling method, an *oracle* labeler, that, for each alarm, yields its true label and a 99% confidence. The oracle bounds represent the result of the approach if we were able to perfectly label the alarms dataset, eliminating the labeling and confidence uncertainty. Oracle bounds always contain the true curve and are sufficiently wide enough to account for sampling uncertainty; therefore, we seek bounds that are at least as wide as the oracle bounds. Table 2 summarizes the average width of the sensitivity and specificity bounds as well as the percentage of the true curve contained in the sensitivity, specificity, and tradeoff confidence bounds. We defer to Appendix C for the hyperparameters that produce the confidence bounds in the aforementioned table and figures. We note that only a subset of the thresholds for each alarm classifier can plausibly be adopted into a clinical setting, and hence only portions of these curves are clinically relevant. In discussion with pediatric physicians, we defined the following clinically relevant regions: 80 to 95 for low SpO_2, 0 to 30 for low RR, 60 to 120 for high RR, 50 to 120 for low HR, and 160 to 220 for high HR.

Our main observation is that our approach produces confidence bounds with *high containment*. Furthermore, the results demonstrate that bound containment improves as progressively more accurate and data efficient probabilistic labeling methods are used. Our confidence uncertainty-aware approach yields 100% containment for 11, 11, and 15 out of 15 bounds using majority vote, uninformed, and informed generative models, respectively. While most majority vote bounds have high containment, we can understand the instances where it is low by looking at the low SpO_2 case study tradeoff bounds (Figure 5, top row). The bounds bow outward unlike traditional tradeoff curves, and if we flipped the labels it would bow inward and improve containment. This implies that the majority vote labeled incorrectly with high-confidence as we anticipated in Section 5.3. The uninformed generative model is disadvantaged when the actual class balance is heavily skewed. This can explain the lower containment in the high HR and the low SpO_2 case. Finally, since *all bounds* produced by an informed generative model achieve 100% containment of the true curve, our approach has effectively captured this outcome of a hypothetical observational study.

[3]Code and data is available at https://github.com/sfpugh/Evaluating-Alarm-Classifiers-with-High-Confidence-Data-Programming.
[4]The smallest possible label confidence is 50%, hence $\epsilon_j = 0.5$ uses all of our data.

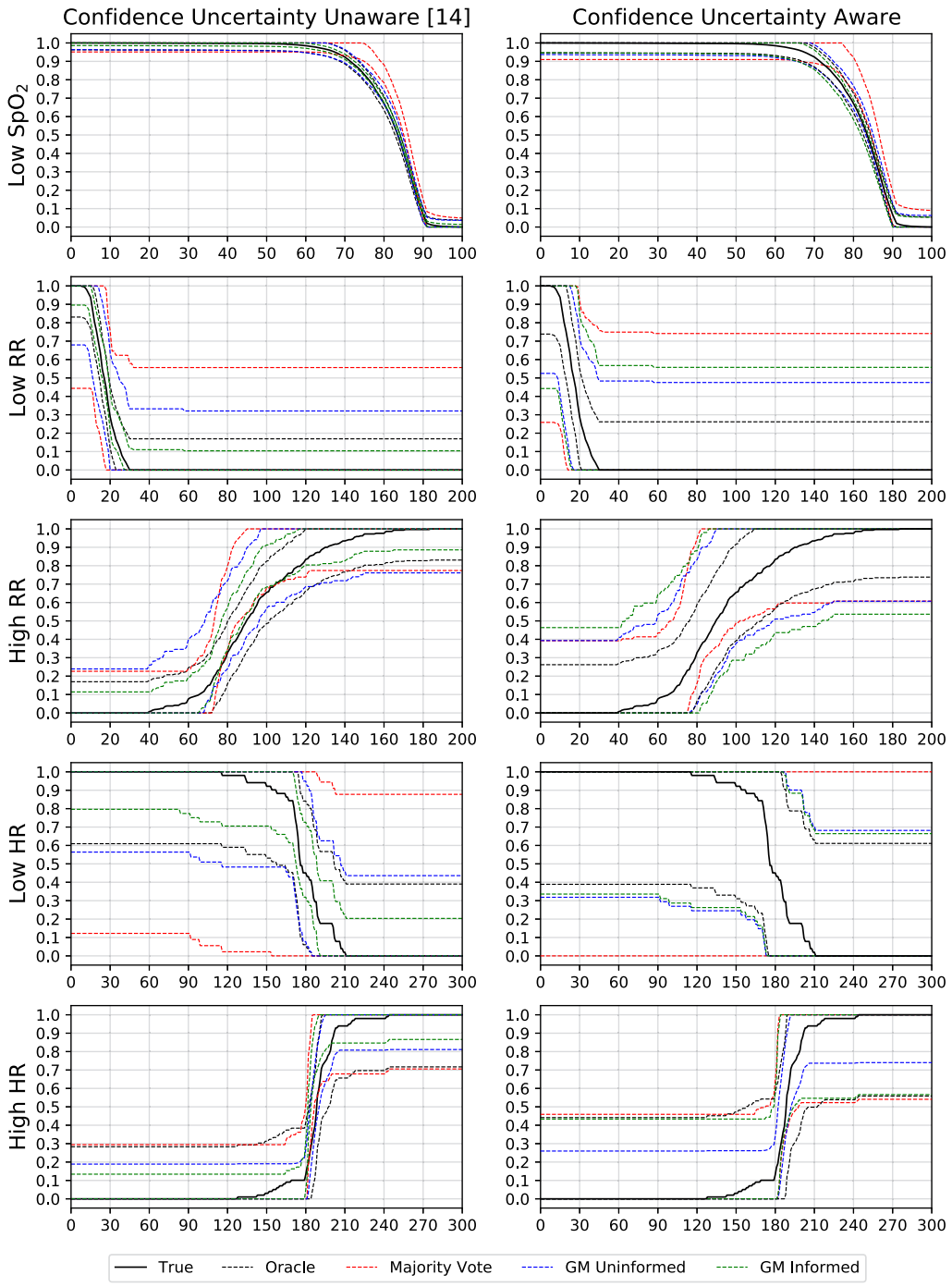

Fig. 3. Sensitivity confidence bounds.

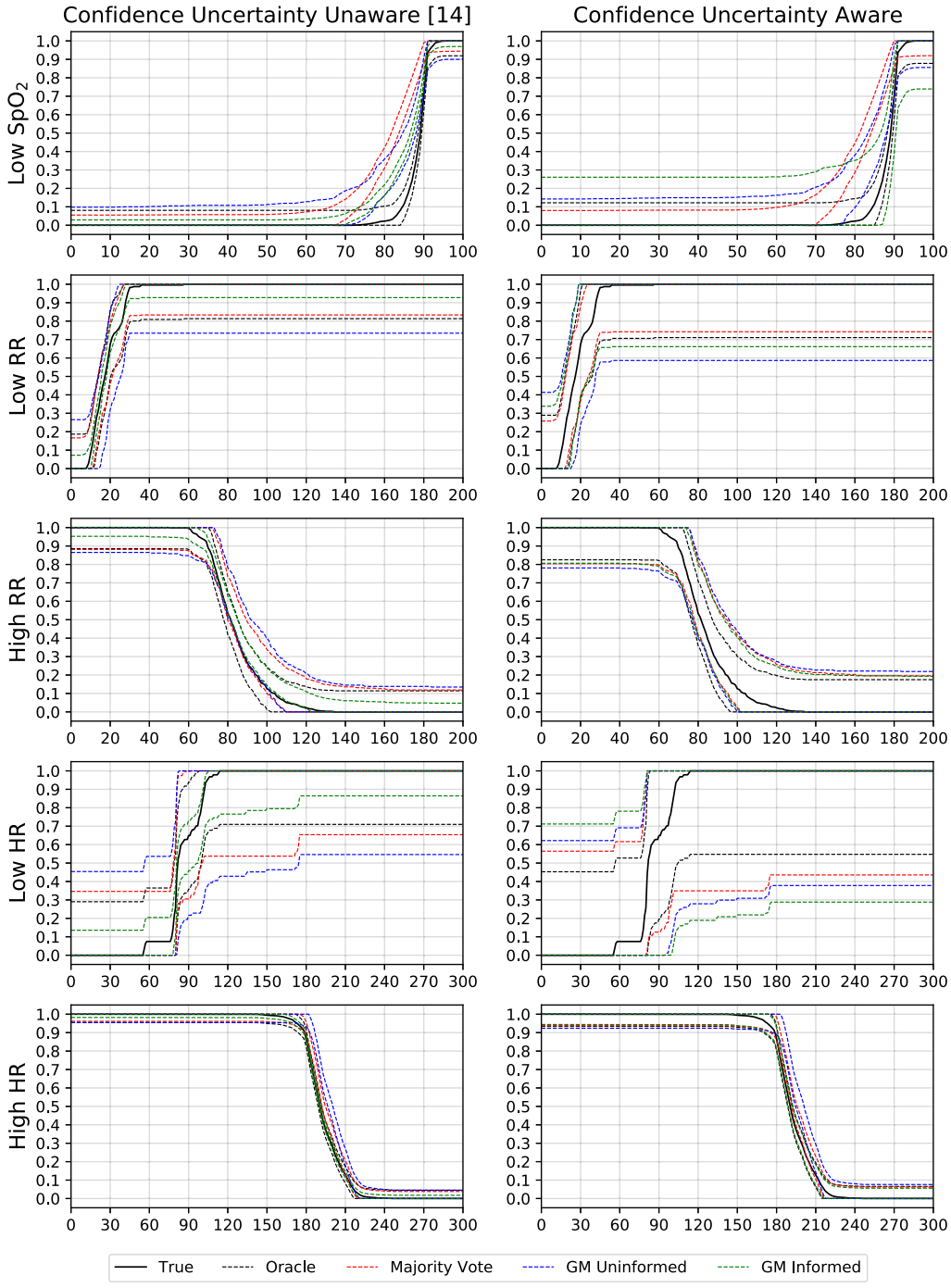

Fig. 4. Specificity confidence bounds.

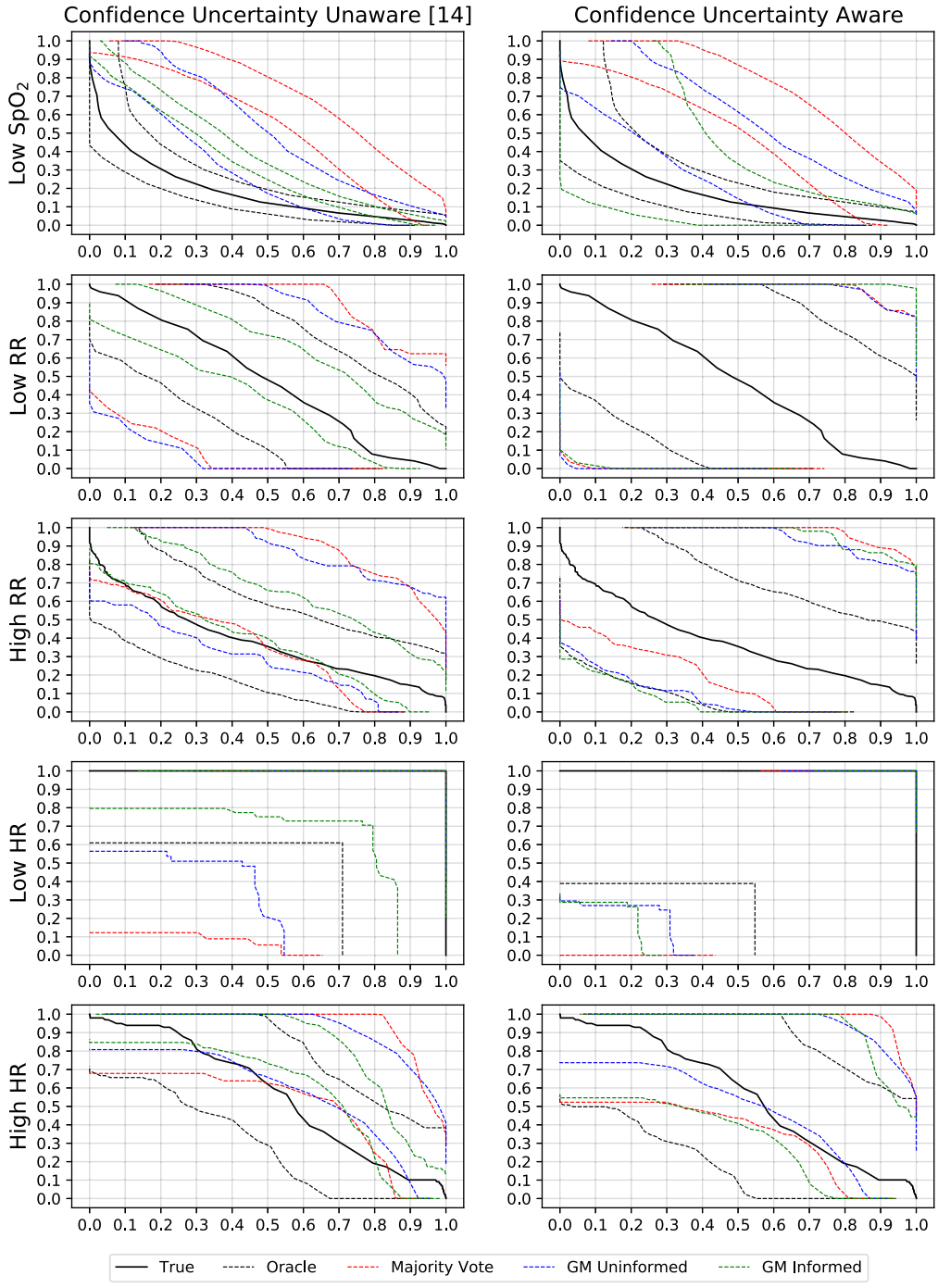

Fig. 5. Tradeoff confidence bounds.

Table 2. Average Confidence Bound Width and Percentage of the True Curve Contained in the Confidence Bounds for Significance Level $p_0 = p_1 = 0.1$

Case Study	Rate	Metric	Confidence Uncertainty Unaware [14]				Confidence Uncertainty Aware			
			Majority Vote	GM Uninformed	GM Informed	Oracle	Majority Vote	GM Uninformed	GM Informed	Oracle
Low SpO_2	Sensitivity	Width	**0.060**	0.049	0.020	0.050	0.107	0.082	**0.071**	0.070
		Containment	0.519	0.897	0.683	-	0.641	**1.000**	**1.000**	-
	Specificity	Width	0.072	**0.129**	0.037	0.089	0.102	**0.176**	0.289	0.133
		Containment	0.562	0.583	0.572	-	0.572	0.727	**1.000**	-
	Tradeoff	Containment	0.115	0.350	0.200	-	0.194	0.483	**1.000**	-
Low RR	Sensitivity	Width	0.572	**0.345**	0.115	0.183	0.757	**0.500**	0.584	0.280
		Containment	**1.000**	**1.000**	**1.000**	-	**1.000**	**1.000**	**1.000**	-
	Specificity	Width	0.182	**0.286**	0.080	0.204	0.279	0.439	**0.360**	0.313
		Containment	**1.000**	**1.000**	**1.000**	-	**1.000**	**1.000**	**1.000**	-
	Tradeoff	Containment	**1.000**	**1.000**	**1.000**	-	**1.000**	**1.000**	**1.000**	-
High RR	Sensitivity	Width	**0.265**	0.314	0.161	0.235	**0.448**	0.488	0.568	0.347
		Containment	0.732	**1.000**	0.687	-	**1.000**	**1.000**	**1.000**	-
	Specificity	Width	**0.154**	0.173	0.066	0.140	0.243	0.268	**0.239**	0.209
		Containment	**1.000**	0.789	0.708	-	**1.000**	**1.000**	**1.000**	-
	Tradeoff	Containment	0.617	**1.000**	0.488	-	**1.000**	**1.000**	**1.000**	-
Low HR	Sensitivity	Width	0.912	**0.488**	0.252	0.433	1.000	0.732	**0.715**	0.654
		Containment	**1.000**	**1.000**	**1.000**	-	**1.000**	**1.000**	**1.000**	-
	Specificity	Width	**0.397**	0.512	0.173	0.318	**0.611**	0.676	0.760	0.485
		Containment	**1.000**	**1.000**	**1.000**	-	**1.000**	**1.000**	**1.000**	-
	Tradeoff	Containment	**1.000**	**1.000**	**1.000**	-	**1.000**	**1.000**	**1.000**	-
High HR	Sensitivity	Width	0.312	0.202	0.146	0.314	**0.476**	0.276	0.449	0.475
		Containment	0.854	**1.000**	0.700	-	**1.000**	**1.000**	**1.000**	-
	Specificity	Width	0.046	0.053	0.022	0.054	0.073	**0.088**	0.066	0.077
		Containment	0.572	0.500	**1.000**	-	0.662	0.581	**1.000**	-
	Tradeoff	Containment	0.629	0.410	0.463	-	**1.000**	0.761	**1.000**	-

For each approach, we put the bound width that is closest to and at least as wide as the oracle bounds in bold face. We also put the best (largest) true curve containment out of both approaches and all probabilistic labeling methods in bold face.

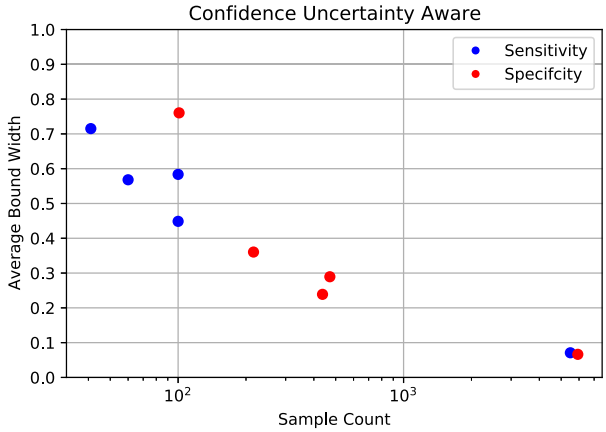

Fig. 6. Sample count (log) versus the average confidence bound width using our confidence uncertainty-aware approach for each case study.

By comparing the left and the right columns in Figures 3–5, we conclude that accounting for confidence uncertainty trades *bound width for higher containment*. The confidence uncertainty-unaware approach yields 100% containment for 7, 9, and 7 out of 15 bounds using majority vote, uninformed, and informed generative models, respectively. In the cases where 100% containment is not achieved, the confidence bounds produced by our approach have better containment. On average, the improvement is 0.187 ± 0.12, 0.224 ± 0.17, and 0.437 ± 0.16 for each labeling method, and 0.284 ± 0.19 overall. However, our approach's bounds are wider than the confidence uncertainty-unaware bounds in all cases by 0.178 ± 0.14 on average. Confidence uncertainty widens the bounds in order to capture more of the hypothetical observational study result.

We compare each approach to their respective oracle bounds and find that the confidence uncertainty-aware approach yields bounds that are *good approximations of the oracle bounds*. Our approach's bounds are wider than the oracle bounds in 24 out of 30 cases by 0.129 ± 0.12 on average. The confidence uncertainty-unaware approach's bounds are wider than the oracle bounds in 13 out of 30 cases by 0.127 ± 0.14 on average. Our uncertainty-aware bounds better account for the sampling uncertainty than the uncertainty-unaware approach. Furthermore, our approach using an informed generative model is able to achieve the same containment as the oracle bounds (100% containment) in all cases, whereas the confidence uncertainty-unaware approach achieves oracle containment in less than half of the cases. Hence, the bounds produced by our approach, though slightly wider, converge to the quality of the best possible confidence bounds producible by our approach.

The width of our bounds *varies appropriately* with the amount of provided data. For example, the bounds for larger dataset cases, low SpO_2, high RR, and high HR, are of moderate width. However, small dataset cases, low RR and low HR, have wide bounds. Bound width is directly affected by the number of samples used (determined by the dataset $\mathcal{I}$ and label uncertainty ϵ_j) and the choice of significance level p_j (which follows from Corollary 1). We explore the effect of the sample count per class in the dataset on the bound width using our approach with an informed generative model in Figure 6. This figure shows that having more data yields tighter bounds.

Recall the context of the case study is to better pick alarm thresholds of a previously poorly calibrated alarm-generating device. In practice, to help determine the best threshold for alarm suppression classifiers, hospital policy makers could use the sensitivity/specificity tradeoff. A good threshold would produce specificity close to 1 (i.e., not suppress any non-suppressible alarms) while maximizing sensitivity (i.e., silence as many suppressible alarms as possible). Suppose policy makers decide to allow a minimum of 90% specificity. For the low vital sign alarms, the minimum threshold using our approach using an informed generative model and the true curve, respectively, are 92 and 91 for SpO_2, 27 and 28 for RR, and 114 and 103 for HR. For the high vital sign alarms,

the respective maximum thresholds are 70 and 69 for RR, and 178 and 178 for HR. The thresholds selected using our approach are sufficiently accurate approximations, if not exact, to those that would have been selected in an observational study.

Limitations: Our confidence bounds are accurate when suppression accuracy is relatively consistent on different high-confidence labels, as stated in Assumption 1. This assumption may be violated in contexts with few available samples or when high-confidence labeling is particularly biased/inaccurate — and then our theoretical guarantees might not hold. For example, there may exist a subset of the patient population for which the clinician-designed labeling functions always label incorrectly. Alternatively, the high-confidence datasets may be biased by only containing alarms of a patient subset not representative of the entire patient population. Our conclusions are expected to generalize to the setting of our data collection, i.e., actionable alarms for pediatric medical-floor patients. To apply our method in a different setting, one may need to elicit different/more labeling functions, and so the tightness and accuracy of the confidence bounds may vary.

8 CONCLUSION

In this article, we proposed an approach for estimating the performance of a clinical alarm classification system without access to labeled alarm data. Probabilistic labels obtained via generative modeling were used as a proxy for the unknown true labels to estimate the average true rates. We then quantified confidence bounds for these estimates. Finally, we evaluated our approach in case studies of low SpO_2, low/high respiratory rate, and low/high heart rate alarms. The results show that we outperform prior work and produce confidence bounds that contain most, if not all, of the true curves generated in an observational study. For future work, we plan to analyze the impact of the number of weak labeling functions used in our approach, automate the extraction of weak labeling functions that satisfy the assumptions of generative models, as well as explore unsupervised confidence calibration of generative models for data programming.

APPENDICES

A THEOREM AND COROLLARY PROOF

We start with an unlabeled dataset of alarms $\mathcal{I}$ and a hypothetical, unavailable dataset of alarms $\mathcal{I}^*$. We pick some subset of high-confidence samples:

$$\mathcal{I}_j \subseteq \{x \in \mathcal{I} \mid f(x) = j \wedge g(x) \geq 1 - \epsilon_j\}.$$

Recall our assumption of consistent true rates:

$$\left|\mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j^*\right] - \mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j\right]\right| \leq 1 - \mathbb{E}\left[g(\tilde{x}) \mid \tilde{x} \in \mathcal{I}_j\right]$$

Then,

$$\begin{aligned}
&\left|\mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j^*\right] - \mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j\right]\right| \leq 1 - \mathbb{E}\left[g(\tilde{x}) \mid \tilde{x} \in \mathcal{I}_j\right] \\
\implies &\Bigg|\mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j^*\right] - \frac{1}{|\mathcal{I}_j^*|}\sum_{x^* \in \mathcal{I}_j^*} \mathbb{1}\left(S(x^*) = j\right) \\
&\quad + \frac{1}{|\mathcal{I}_j^*|}\sum_{x^* \in \mathcal{I}_j^*} \mathbb{1}\left(S(x^*) = j\right) - \frac{1}{|\mathcal{I}_j|}\sum_{x \in \mathcal{I}_j} \mathbb{1}\left(S(x) = j\right) \\
&\quad + \frac{1}{|\mathcal{I}_j|}\sum_{x \in \mathcal{I}_j} \mathbb{1}\left(S(x) = j\right) - \mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j\right]\Bigg|
\end{aligned}$$

$$
\begin{aligned}
&\leq 1 - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} g(x) + \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} g(x) - \mathbb{E}\left[g(\tilde{x}) \mid \tilde{x}\in\mathcal{I}_j\right] \\
&\implies \left|\frac{1}{|\mathcal{I}_j^*|}\sum_{x^*\in\mathcal{I}_j^*} \mathbb{1}\left(S(x^*)=j\right) - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} \mathbb{1}\left(S(x)=j\right)\right| \\
&\left|\mathbb{E}\left[S(\tilde{x})=j \mid \tilde{x}\in\mathcal{I}_j^*\right] - \frac{1}{|\mathcal{I}_j^*|}\sum_{x^*\in\mathcal{I}_j^*} \mathbb{1}\left(S(x^*)=j\right)\right| \\
&\left|\mathbb{E}\left[S(\tilde{x})=j \mid \tilde{x}\in\mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} \mathbb{1}\left(S(x)=j\right)\right| \\
&\leq 1 - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} g(x) + \left(\frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} g(x) - \mathbb{E}\left[g(\tilde{x}) \mid \tilde{x}\in\mathcal{I}_j\right]\right) \\
&\implies \left|\frac{1}{|\mathcal{I}_j^*|}\sum_{x^*\in\mathcal{I}_j^*} \mathbb{1}\left(S(x^*)=j\right) - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} \mathbb{1}\left(S(x)=j\right)\right| \\
&\leq 1 - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} g(x) \\
&\quad + \left|\mathbb{E}\left[S(\tilde{x})=j \mid \tilde{x}\in\mathcal{I}_j^*\right] - \frac{1}{|\mathcal{I}_j^*|}\sum_{x^*\in\mathcal{I}_j^*} \mathbb{1}\left(S(x^*)=j\right)\right| \\
&\quad + \left|\mathbb{E}\left[S(\tilde{x})=j \mid \tilde{x}\in\mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} \mathbb{1}\left(S(x)=j\right)\right| \\
&\quad + \left|\mathbb{E}\left[g(\tilde{x}) \mid \tilde{x}\in\mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} g(x)\right|
\end{aligned}
$$

Let $\delta_j = 1 - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} g(x)$ and $\gamma_j = \frac{(c_j-\delta_j)\sqrt{|\mathcal{I}_j^*|}}{\sqrt{|\mathcal{I}_j|}+2\sqrt{|\mathcal{I}_j^*|}}$. Then for any alarm classifier S it holds that:

$$
\begin{aligned}
&\mathbb{P}\left[\left|R_j(\mathcal{I}_j) - R_j(\mathcal{I}_j^*)\right| \geq c_j\right] \\
&= \mathbb{P}\left[\left|\frac{1}{|\mathcal{I}_j^*|}\sum_{x^*\in\mathcal{I}_j^*} \mathbb{1}\left(S(x^*)=j\right) - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} \mathbb{1}\left(S(x)=j\right)\right| \geq c_j\right] \\
&\leq \mathbb{P}\left[\delta_j + \left|\mathbb{E}\left[S(\tilde{x})=j \mid \tilde{x}\in\mathcal{I}_j^*\right] - \frac{1}{|\mathcal{I}_j^*|}\sum_{x^*\in\mathcal{I}_j^*} \mathbb{1}\left(S(x^*)=j\right)\right|\right. \\
&\quad + \left|\mathbb{E}\left[S(\tilde{x})=j \mid \tilde{x}\in\mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} \mathbb{1}\left(S(x)=j\right)\right| \\
&\quad \left. + \left|\mathbb{E}\left[g(\tilde{x}) \mid \tilde{x}\in\mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x\in\mathcal{I}_j} g(x)\right| \geq c_j\right]
\end{aligned}
$$

$$
\begin{aligned}
&= \mathbb{P}\Bigg[\Bigg|\mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j^*\right] - \frac{1}{|\mathcal{I}_j^*|}\sum_{x^* \in \mathcal{I}_j^*} \mathbb{1}\left(S(x^*) = j\right)\Bigg| \\
&\quad + \Bigg|\mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x \in \mathcal{I}_j} \mathbb{1}\left(S(x) = j\right)\Bigg| \\
&\quad + \Bigg|\mathbb{E}\left[g(\tilde{x}) \mid \tilde{x} \in \mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x \in \mathcal{I}_j} g(x)\Bigg| \geq (c_j - \delta_j - 2\gamma_j) + \gamma_j + \gamma_j\Bigg] \\
&\leq \mathbb{P}\Bigg[\Bigg|\mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j^*\right] - \frac{1}{|\mathcal{I}_j^*|}\sum_{x^* \in \mathcal{I}_j^*} \mathbb{1}\left(S(x^*) = j\right)\Bigg| \geq c_j - \delta_j - 2\gamma_j\Bigg] \\
&\quad + \mathbb{P}\Bigg[\Bigg|\mathbb{E}\left[S(\tilde{x}) = j \mid \tilde{x} \in \mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x \in \mathcal{I}_j} \mathbb{1}\left(S(x) = j\right)\Bigg| \geq \gamma_j\Bigg] \\
&\quad + \mathbb{P}\Bigg[\Bigg|\mathbb{E}\left[g(\tilde{x}) \mid \tilde{x} \in \mathcal{I}_j\right] - \frac{1}{|\mathcal{I}_j|}\sum_{x \in \mathcal{I}_j} g(x)\Bigg| \geq \gamma_j\Bigg] \\
&\leq 2\exp\left(-2|\mathcal{I}_j^*|(c_j - \delta_j - 2\gamma_j)^2\right) + 4\exp\left(-2|\mathcal{I}_j|\gamma_j^2\right) \\
&\leq 6\exp\left(\frac{-2|\mathcal{I}_j||\mathcal{I}_j^*|}{\left(\sqrt{|\mathcal{I}_j|} + 2\sqrt{|\mathcal{I}_j^*|}\right)^2}(c_j - \delta_j)^2\right).
\end{aligned}
$$

For the corollary, we are given desired significance level p_j:

$$
p_j = 6\exp\left(\frac{-2|\mathcal{I}_j||\mathcal{I}_j^*|}{\left(\sqrt{|\mathcal{I}_j|} + 2\sqrt{|\mathcal{I}_j^*|}\right)^2}(c_j + \eta_j - 1)^2\right).
$$

Solving the equation above for c_j yields the following equation for the bound size,

$$
c_j = 1 - \eta_j + \sqrt{\frac{\left(\sqrt{|\mathcal{I}_j|} + 2\sqrt{|\mathcal{I}_j^*|}\right)^2}{-2|\mathcal{I}_j||\mathcal{I}_j^*|}\ln\left(\frac{p_j}{6}\right)}.
$$

B LABELING FUNCTIONS

The twelve guidelines for deciding whether an alarm is suppressible or non-suppressible from Section 5.2 are encoded as **labeling functions** (**LFs**) in the following ways.

(1) *LF_long_alarm_T* labels non-suppressible if the length of the alarm is at least T seconds, otherwise it abstains. LFs 1 through 3 use $T = 60, 65,$ and 70, respectively.
(2) *LF_spo2_aboveX_belowY_overT* labels non-suppressible if SpO_2 stays within range $(X, Y]$ for longer than T seconds after the alarm start, otherwise it abstains. LFs 4 to 9 use $(X, Y, T) = (80, 85, 120), (0, 80, 120), (70, 80, 100), (60, 70, 90), (50, 60, 60),$ and $(0, 50, 30)$, respectively.
(3) *LF_hr_aboveX_overT* labels non-suppressible if heart rate is above X for longer than T seconds after the alarm start, otherwise it abstains. LF 10 uses $X = 220$ and $T = 10$.

(4) *LF_hr_aboveX_belowY_overT* labels non-suppressible if heart rate stays withing range $(X, Y]$ for longer than T seconds after the alarm start, otherwise it abstains. LFs 11 to 14 use $(X, Y, T) = (0, 50, 10)$, $(40 \cdot \alpha, 50 \cdot \alpha, 120)$, $(30 \cdot \alpha, 40 \cdot \alpha, 60)$, and $(0, 30 \cdot \alpha, 0)$, respectively, where α is a scaling age factor taking value of 3.833 for less than one month, 3.766 for one month to less than two months, 3.733 for two months to less than six months, 3.533 for six months and older.
(5) *LF_rr_aboveX_belowY_overT* labels non-suppressible if respiratory rate stays within range $(X, Y]$ for longer than T seconds after the alarm start, otherwise it abstains. LFs 15 to 18 use $(X, Y, T) = (0, 10, 120)$, $(40 \cdot \alpha, 50 \cdot \alpha, 120)$, $(30 \cdot \alpha, 40 \cdot \alpha, 60)$, and $(0, 30 \cdot \alpha, 0)$, respectively, where α is a scaling age factor taking value of 0.933 for less than one month, 0.900 for one month to less than two months, 0.866 for two months to less than six months, 0.800 for six months and older.
(6) *LF_repeat_Xalarms_inT* labels non-suppressible if at least X additional low SpO_2 alarms sound within T seconds of the alarm, otherwise it abstains. LFs 19 to 22 use $(X, T) = (1, 15)$, $(1, 30)$, $(1, 60)$, and $(10, 300)$, respectively.
(7) *LF_short_alarm_T* labels suppressible if the length of the alarm is at most T seconds, otherwise it abstains. LFs 23 to 25 use $T = 5, 10,$ and 15, respectively.
(8) *LF_recoverX_inT* labels suppressible if SpO_2 improves by more than X percent within T seconds of the alarm, otherwise it abstains. LFs 26 and 27 use $(X, T) = (20, 10)$ and $(20, 15)$, respectively.
(9) *LF_hr_tech_error_X* labels suppressible if the absolute difference between ECG heart rate and pulse oximeter heart rate is greater than X at the time of alarm, otherwise it abstains. LFs 28 and 29 use $X = 20$ and 30 respectively.
(10) *LF_bad_spo2_waveform_X_T* labels suppressible if there exists an outlier with value greater than X within a T second window of the alarm in the SpO_2 waveform matrix profile, otherwise it abstains.[5] LFs 30 to 40 use $(X, T) = (8.4, 120)$, $(7.8, 110)$, $(7.2, 100)$, $(6.6, 90)$, $(6.0, 80)$, $(5.3, 70)$, $(4.6, 60)$, $(3.8, 50)$, $(2.9, 40)$, $(2.1, 30)$, and $(1.0, 20)$, respectively.
(11) *LF_bad_hr_waveform_X_T* labels suppressible if there exists an outlier with value greater than X within a T second window of the alarm in the heart rate waveform matrix profile, otherwise it abstains. LFs 41 to 51 use $(X, T) = (9.0, 120)$, $(8.5, 110)$, $(7.8, 100)$, $(7.3, 90)$, $(6.7, 80)$, $(6.0, 70)$, $(5.4, 60)$, $(4.7, 50)$, $(3.9, 40)$, $(3.1, 30)$, and $(2.1, 20)$, respectively.
(12) *LF_bad_rr_waveform_X_T* labels suppressible if there exists an outlier with value greater than X within a T second window of the alarm in the respiratory rate waveform matrix profile, otherwise it abstains. LFs 52 to 62 use $(X, T) = (8.7, 120)$, $(8.1, 110)$, $(7.6, 100)$, $(7.1, 90)$, $(6.5, 80)$, $(6.0, 70)$, $(5.4, 60)$, $(4.7, 50)$, $(3.9, 40)$, $(3.0, 30)$, and $(2.0, 20)$, respectively.

C ADDITIONAL RESULTS

Table C.1 summarizes the hyperparameters that produce the confidence bounds in Figures 3–5. Recall that ϵ_j is the allowed level of label uncertainty in both approaches, and γ_j is an admissable free parameter that applies only to the confidence uncertainty unaware approach [14].

[5] To find anomalies in vitals waveforms we analyze their matrix profiles. At a high-level, a matrix profile represents the dissimilarity between each vital sign measurement in the data and the rest of the data. Large values in the matrix profile correspond to outliers. We use the matrixprofile-ts Python library (github.com/matrix-profile-foundation/matrixprofile).

Table C.1. Summary of Hyperparameters Corresponding to the Confidence Bounds in Figures 3–5

Case Study	Hyperparameter	Confidence Uncertainty Unaware [14]			Confidence Uncertainty Aware		
		Majority Vote	GM Uninformed	GM Informed	Majority Vote	GM Uninformed	GM Informed
Low SpO_2	ϵ_0	0.001	0.041	0.495	0.001	0.067	0.155
	ϵ_1	0.001	0.001	0.497	0.001	0.001	0.016
	γ_0	0.021	0.021	0.360	-	-	-
	γ_1	0.034	0.034	0.138	-	-	-
Low RR	ϵ_0	0.001	0.094	0.499	0.001	0.177	0.140
	ϵ_1	0.334	0.163	0.491	0.500	0.246	0.320
	γ_0	0.079	0.079	0.332	-	-	-
	γ_1	0.191	0.556	0.460	-	-	-
High RR	ϵ_0	0.001	0.041	0.495	0.001	0.091	0.053
	ϵ_1	0.001	0.086	0.466	0.125	0.156	0.024
	γ_0	0.066	0.066	0.224	-	-	-
	γ_1	0.142	0.142	0.419	-	-	-
Low HR	ϵ_0	0.001	0.233	0.344	0.200	0.323	0.344
	ϵ_1	0.500	0.110	0.383	0.500	0.141	0.064
	γ_0	0.201	0.175	0.555	-	-	-
	γ_1	0.243	0.886	0.437	-	-	-
High HR	ϵ_0	0.001	0.009	0.029	0.001	0.030	0.019
	ϵ_1	0.001	0.001	0.288	0.143	0.001	0.014
	γ_0	0.020	0.020	0.036	-	-	-
	γ_1	0.156	0.156	0.282	-	-	-

REFERENCES

[1] Wan-Tai M. Au-Yeung, Ashish K. Sahani, Eric M. Isselbacher, and Antonis A. Armoundas. 2019. Reduction of false alarms in the intensive care unit using an optimized machine learning based approach. *NPJ Digital Medicine* 2, 1 (2019), 1–5.

[2] S. Balaswamy and R. Vishnu Vardhan. 2015. Confidence interval estimation of an ROC curve: An application of Generalized Half Normal and Weibull distributions. *Journal of Probability and Statistics* 2015 (2015).

[3] Christopher P. Bonafide, A. Russell Localio, John H. Holmes, Vinay M. Nadkarni, Shannon Stemler, Matthew MacMurchy, Miriam Zander, Kathryn E. Roberts, Richard Lin, and Ron Keren. 2017. Video analysis of factors associated with response time to physiologic monitor alarms in a children's hospital. *JAMIA Pediatrics* 171, 1 (2017), 524–531.

[4] Corinna Cortes and Mehryar Mohri. 2004. Confidence intervals for the area under the ROC curve. *Advances in Neural Information Processing Systems* 17 (2004).

[5] Barbara J. Drew, Patricia Harris, Jessica K. Zègre-Hemsey, Tina Mammone, Daniel Schindler, Rebeca Salas-Boni, Yong Bai, Adelita Tinoco, Quan Ding, and Xiao Hu. 2014. Insights into the problem of alarm fatigue with physiologic monitor devices: A comprehensive observational study of consecutive intensive care unit patients. *PloS one* 9, 10 (2014), e110274.

[6] Kuk Jang, James Weimer, Houssam Abbas, Zhihao Jiang, Jackson Liang, Sanjay Dixit, and Rahul Mangharam. 2018. Computer aided clinical trials for implantaule cardiac devices. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2018), 1–4.

[7] Leo Kobayashi, John W. Gosbee, and Derek L. Merck. 2017. Development and application of a clinical microsystem simulation methodology for human factors-based research of alarm fatigue. *HERD: Health Environments Research and Design Journal* 10, 4 (2017), 91–104.

[8] Petre Lameski, Eftim Zdravevski, Saso Koceski, Andrea Kulakov, and Vladimir Trajkovik. 2017. Suppression of intensive care unit false alarms based on the arterial blood pressure signal. *IEEE Access* 5, 4 (2017), 5829–5836.

[9] Matt MacMurchy, Shannon Stemler, Mimi Zander, and Christopher P. Bonafide. 2017. Research: Acceptability, feasibility, and cost of using video to evaluate alarm fatigue. *Biomedical Instrumentation and Technology* 51, 1 (2017), 25–33.
[10] Hung Nguyen, Sooyong Jang, Radoslav Ivanov, Christopher Bonafide, James Weimer, and Insup Lee. 2018. Reducing pulse oximetry false alarms without missing life-threatening events. *Smart Health* 9–10 (2018), 287–296.
[11] Christine Weirich Paine, Veena V. Goel, Elizabeth Ely, Christopher D. Stave, Shannon Stemler, Miriam Zander, and Christopher P. Bonafide. 2016. Systematic review of physiologic monitor alarm characteristics and pragmatic interventions to reduce alarm frequency. *Journal of Hospital Medicine* 11, 2 (2016), 136–144.
[12] Arghya Pal and Vineeth N. Balasubramanian. 2018. Adversarial data programming: Using gans to relax the bottleneck of curated labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1556–1565.
[13] Arghya Pal and Vineeth N. Balasubramanian. 2020. Generative adversarial data programming. arXiv:2005.00364. Retrieved from https://arxiv.org/abs/2005.00364.
[14] Sydney Pugh, Ivan Ruchkin, Christopher Bonafide, Sara DeMauro, Oleg Sokolsky, Insup lee, and James Weimer. 2021. High-confidence data programming for evaluating suppression of physiological alarms. In *Proceedings of the Conference on Connected Health: Applications, Systems and Engineering Technologies*.
[15] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2020. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal* 29, 2 (2020), 709–730.
[16] Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. *Advances in Neural Information Processing Systems* 29 (2016), 3567.
[17] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2018. Training Complex Models with Multi-Task Weak Supervision. arXiv:1810.02840. Retrieved from https://arxiv.org/abs/1810.02840.
[18] Bradford D. Winters, Maria M. Cvach, Christopher P. Bonafide, Xiao Hu, Avinash Konkani, Michael F. O'Connor, Jeffrey M. Rothschild, Nicholas M. Selby, Michele M. Pelter, Barbara McLean, et al. 2018. Technological distractions (part 2): A summary of approaches to manage clinical alarms with intent to reduce alarm fatigue. *Critical Care Medicine* 46, 1 (2018), 130–137. [Accessed January 10, 2022].

Received 26 January 2022; revised 13 June 2022; accepted 14 June 2022