Detection of Acute Respiratory Distress Syndrome by Incorporation of Label Uncertainty and Partially Available Privileged Information*

Elyas Sabeti[1], Joshua Drews[1], Narathip Reamaroon[1], Jonathan Gryak[1], Michael Sjoding[2] and Kayvan Najarian[3]

***Abstract*— Acute respiratory distress syndrome (ARDS) is a fulminant inflammatory lung injury that develops in patients with critical illnesses including sepsis, pneumonia, and trauma. However, many patients with ARDS are not recognized when they develop this syndrome nor given outcome-improving treatments. Because ARDS is a clinical syndrome, physicians may not be certain about a patient's diagnosis (label uncertainty). In addition, the diagnosis requires a chest x-ray, which may not be always be available in a clinical setting (privileged information). For this paper, we implemented the Learning Using Label Uncertainty and Partially Available Privileged Information (LULUPAPI) paradigm, built on classical SVM, to detect ARDS using Electronic Health Record (EHR) data and chest radiography. In comparison to SVM, this resulted in a 3.55 percent improvement of test AUC.**

I. INTRODUCTION

200,000 patients in the United States each year suffer from Acute Respiratory Distress Syndrome (ARDS), a fulminant lung injury. Patients with ARDS have a mortality rate of 30-40% [1]. Simple interventions such as reducing ventilator tidal volume have been shown to improve patient outcomes [2]. However, physician recognition of ARDS ranges from 50 to 80% depending on the severity of condition; as a consequence, many patients do not receive these life-saving treatments [3]. One potentially effective, yet underutilized, method of assisting physicians in recognition of ARDS is the analysis of electronic health record (EHR) data. Algorithms that process information provided by EHR data and bedside monitoring devices can flag patients at risk for ARDS and prompt clinicians to administer treatment.

Such an algorithm presents three particular problems. First, physicians may be equivocal in their diagnoses of ARDS for some patients, and labels present in the training set given for the diagnosis of ARDS by physician experts may be incorrect. Unaccounted for, these incorrect labels will corrupt the trained model. This problem is referred to as Label Uncertainty (LU). Second, chest radiographs are necessary to diagnose ARDS, and therefore chest x-ray results are typically present in the training set. However, an ideal algorithm developed to assist physicians with the diagnosis of ARDS should not depend on chest x-ray results, as they may not always be available at the time of ARDS development. Yet, the radiographs of the training set contain information that may be helpful in classifying patients, so it is desirable to use the radiographs only *to train* the algorithm. In this case, the radiographs are referred to as privileged information (PI), and this problem as Learning Using Privileged Information (LUPI). Third, the assumption of the availability of privileged information for all the training samples is unrealistic. One could divide the dataset into train and test such that all the training samples have privileged information. However, this would create bias, since radiographs are usually ordered for the patients that already seem unhealthy. Hence, partial availability of privileged information should be taken into consideration. Combining these three problems yields the problem of Learning Using Label Uncertainty and Partially Available Privileged Information (LULUPAPI). While the current analysis focuses on the diagnosis of ARDS, the LULUPAPI problem is present in a wide range of other clinical problems both in the hospital (e.g., sepsis) and in outpatient settings (e.g., depression).

There are many proposed approaches to train with label uncertainty. Frenay et al. classified label noises and proposed algorithms that were either robust to, tolerant of, or cleansed label noise [4]. Natarajan et al. considered label noise as a stochastic, class-conditional process and obtained bounds for empirical risk minimization [5]. Duan et al. divided the data between having a higher noise rate and lower before implementing their own classification algorithm [6]. Vembu et al. leveraged noisy labels to help in optimization by considering labels with higher annotator disagreement as likely to be nearer the decision boundary than others [7].

Learning using privileged information accelerates machine learning by more closely mimicking human teacher-student interactions [8]. In human interactions, the teacher provides the student with additional information specific to each example, such as explanations. This allows the student to learn more information from each example and so learn faster [8]. Learning using privileged information has proven successful in several applications. Sharmanska et al. found that learning using privileged information aided computer vision tasks [9]. Ribeiro et al. found that SVM+, a modification of SVM that leverages privileged information, improved bankruptcy

*This work was partially supported by the National Science Foundation under Grant No. 1722801 and by the National Institutes of Health under Grant NHLBI K01HL136687.

[1]Elyas Sabeti, Joshua Drews, Narathip Reamaroon and Jonathan Gryak are with the Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor {sabeti,joshdr,nreamaro,gryakj}@umich.edu

[2]Michael Sjoding is with the Faculty of Department of Internal Medicine, University of Michigan, Ann Arbor msjoding@med.umich.edu

[3]Kayvan Najarian is with the Faculty of Department of Computational Medicine and Bioinformatics, Department of Emergency Medicine, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor kayvan@umich.edu

prediction compared to regular SVM [10]. Liang et al. modified SVM+ to handle multi-task learning and found that it proved more effective than regular SVM [11].

For this paper, we incorporated label uncertainty with partial availability of privileged information (the LULUPAPI paradigm) to detect patients with ARDS.

II. LULUPAPI

The LULUPAPI model can be implemented based on the classical SVM problem. Given a set of training data

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \quad \mathbf{x}_i \in X,\ y_i \in \{-1, 1\}$$

SVM first maps training data $\mathbf{x} \in X$ into vector (space) $\mathbf{z} \in Z$. It then constructs the optimal separating hyperplane by learning the decision rule $f(\mathbf{z}) = \mathbf{w} \cdot \mathbf{z} + b$ where $\mathbf{w}$ and b are hyperplane parameters and the solution of

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t.}\ \forall\, 1 \le i \le n,\ y_i(\mathbf{w}\cdot\mathbf{z}_i + b) \ge 1 - \xi_i \\ &\forall\, 1 \le i \le n,\ \xi_i \ge 0 \end{aligned} \tag{1}$$

where $C > 0$ is a hyperparameter.

In the LUPI paradigm [12], [13], [14], [15], [16], in addition to the standard training data, $\mathbf{x}_i \in X$, $y_i \in \{-1, 1\}$, a "teacher" provides a "student" with *privileged information*, $\mathbf{x}_i^* \in X^*$, which is only available for the training examples and not for the test examples. Hence, the LUPI model requires triplets $(\mathbf{x}_i, \mathbf{x}_i^*, y_i)$ for training. This problem is known as SVM+. In SVM+, in addition to mapping the data vector $\mathbf{x} \in X$ into vector (space) $\mathbf{z} \in Z$, the privileged information $\mathbf{x}^* \in X^*$ is mapped into vector (space) $\mathbf{z}^* \in Z^*$, and the slack variables ξ_i of SVM are replaced by the correcting function $\varphi(\mathbf{z}^*) = \mathbf{w}^* \cdot \mathbf{z}^* + b^*$. As mentioned earlier, the LUPI paradigm assumes that privileged information is available for all the training data; however, this is not necessarily the case. In many practical applications, privileged information is only available for a fraction of the training data. Therefore, assume the training data is provided as m triplets $(\mathbf{x}_i, \mathbf{x}_i^*, y_i)$ of samples with privileged information and $n - m$ pairs $(\mathbf{x}_i, y_i)$ of samples without privileged information. In order to incorporate label uncertainty, we can allow variability in parameter C through training samples based on label confidence. In other words, since the slack variables ξ_i (or the correcting function) permit some misclassification with penalty parameter C to establish soft-margin decision boundaries, data with high label confidence can be given more weight and influence on the decision boundary. This yields the LULUPAPI paradigm, which requires the training samples

$$\begin{aligned} &(\mathbf{x}_1, \mathbf{x}_1^*, y_1, \pi_1), \ldots, (\mathbf{x}_m, \mathbf{x}_m^*, y_m, \pi_m), (\mathbf{x}_{m+1}, y_{m+1}, \pi_{m+1}), \\ &\qquad (\mathbf{x}_{m+2}, y_{m+2}, \pi_{m+2}), \ldots, (\mathbf{x}_n, y_n, \pi_n) \\ &\mathbf{x}_i \in X,\ \mathbf{x}_i^* \in X^*,\ y_i \in \{-1, 1\},\ \pi_i \in \mathbb{R}, \end{aligned}$$

where π_i is a quantitative measure of uncertainty in the labels. In the most natural formulation, we can consider the correcting function for the samples with privileged information and slack variables for the samples without privileged information. However, as suggested in [13], for the samples with privileged information, replacing slack variables with a smooth correcting function $\varphi(\mathbf{z}^*) = \mathbf{w}^* \cdot \mathbf{z}^* + b^*$ may not always be the best choice. Instead, we can use a mixture of slacks as $\xi_i^{'} = (\mathbf{w}^* \cdot \mathbf{z}_i^* + b^*) + \rho\xi_i^*$ for $1 \le i \le m$ and $\rho \in \mathbb{R}$. Then the formulation of LULUPAPI is

$$\begin{aligned} &\min_{\mathbf{w},b,\xi,\mathbf{w}^*,b^*,\xi^*} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{\gamma}{2}\|\mathbf{w}^*\|_2^2 + C\sum_{i=m+1}^{n}\pi_i\xi_i \\ &\qquad + \rho C^*\sum_{i=1}^{m}\pi_i\xi_i^* + C^*\sum_{i=1}^{m}(\mathbf{w}^*\cdot\mathbf{z}_i^* + b^*) \\ &\text{s.t.}\ \forall\, 1 \le i \le m \quad y_i(\mathbf{w}\cdot\mathbf{z}_i + b) \ge 1 - (\mathbf{w}^*\cdot\mathbf{z}_i^* + b^*) - \xi_i^* \\ &\quad \forall\, 1 \le i \le m \quad \mathbf{w}^*\cdot\mathbf{z}_i^* + b^* \ge 0 \\ &\quad \forall\, 1 \le i \le m \quad \xi_i^* \ge 0 \\ &\forall\, m+1 \le i \le n \quad y_i(\mathbf{w}\cdot\mathbf{z}_i + b) \ge 1 - \xi_i \\ &\forall\, m+1 \le i \le n \quad \xi_i \ge 0 \end{aligned} \tag{2}$$

where $C > 0$, $C^* > 0$ and $\gamma > 0$ are hyperparameters. The term $\frac{\gamma}{2}\|\mathbf{w}^*\|_2^2$ restricts the VC-dimension of the function space. This formulation has many interesting properties, e.g. its performance is always lower bounded by SVM's performance. The dual optimization problem of (2) can be written as

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} D(\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K_{i,j} \tag{3}$$

$$-\frac{1}{2\gamma}\sum_{i,j=1}^{m}(\alpha_i + \beta_i - C^*)(\alpha_j + \beta_j - C^*)K_{i,j}^*$$

$$\text{s.t.} \quad \sum_{i=1}^{n} y_i\alpha_i = 0 \tag{4}$$

$$\sum_{i=1}^{m}(\alpha_i + \beta_i - C^*) = 0 \tag{5}$$

$$\forall\, m+1 \le i \le n, \quad 0 \le \alpha_i \le \pi_i C \tag{6}$$

$$\forall\, 1 \le i \le m, \quad 0 \le \alpha_i \le \rho\pi_i C^*, \quad 0 \le \beta_i \tag{7}$$

where $K_{i,j} \triangleq K(\mathbf{z}_i, \mathbf{z}_j)$ and $K_{i,j}^* \triangleq K^*(\mathbf{z}_i^*, \mathbf{z}_j^*)$ are the kernels in the decision space and the correcting space respectively, with the decision function

$$f(\mathbf{z}) = \mathbf{w}\cdot\mathbf{z} + b = \sum_{i=1}^{n} y_i\alpha_i K(\mathbf{z}_i, \mathbf{z}) + b. \tag{8}$$

In next section, we propose an SMO-style algorithm for LULUPAPI optimization.

III. OPTIMIZATION ALGORITHM

A widely used algorithm for solving the SVM dual problem is sequential minimal optimization (SMO) [17]. An SMO-style algorithm was previously used for LUPI with ubiquitous privileged information [15], [16]. Inspired by these works, we suggest an SMO-style algorithm for solving LULUPAPI. Our proposed algorithm works in an iterative way. The problem of equation (3) can be considered as the general form of

$$\max_{\boldsymbol{\theta}\in\mathcal{F}} D(\boldsymbol{\theta})$$

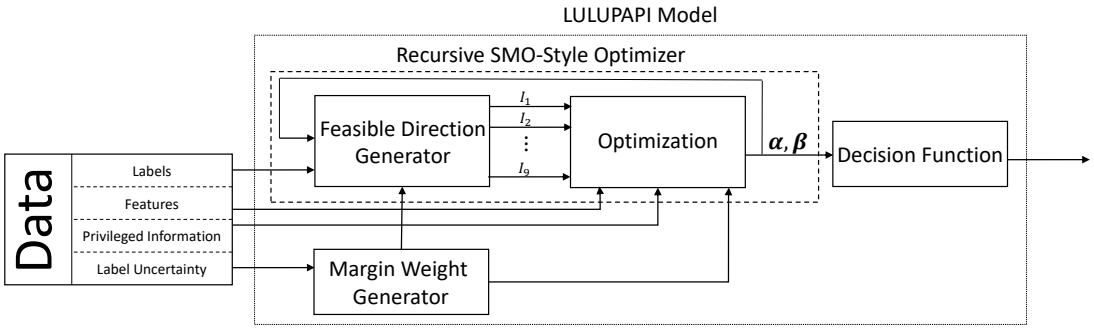

Fig. 1. SMO-style optimizer for the LULUPAPI model

where $\boldsymbol{\theta} \in \mathbb{R}^k$, $D : \mathbb{R}^k \to \mathbb{R}$ is a concave quadratic function, and $\mathcal{F}$ is a convex compact set of linear equalities and inequalities. At each iteration our algorithm finds the *maximally sparse feasible directions* defined as follows:

Definition 1. A direction $\mathbf{u} \in \mathbb{R}^k$ with $n_1 < k$ zero elements is a *maximally sparse feasible direction* at the point $\boldsymbol{\theta} \in \mathcal{F}$ if $\exists \lambda > 0$ such that $\boldsymbol{\theta} + \lambda \mathbf{u} \in \mathcal{F}$ and any $\mathbf{u}_2 \in \mathbb{R}^k$ with $n_2 < k$ zero elements such that $n_1 < n_2$ is not feasible.

These directions are irreducible working sets that satisfy each constraint (4,5,6,7). Among these directions, the algorithm chooses that which maximizes the cost function. The cost function in equation (3) has $n+m$ variables, i.e., $\{\alpha_i\}_{i=1}^n$ and $\{\beta_i\}_{i=1}^m$. We abbreviate $\boldsymbol{\theta} \triangleq (\boldsymbol{\alpha}, \boldsymbol{\beta})^T$ as a $n+m$ vector of variables, the concatenation of the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ variables. Hence each maximally sparse feasible direction is $\mathbf{u} \in \mathbb{R}^{n+m}$. It can be verified that (3) has 9 groups of such directions:

1) $I_1 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2),\ n+1 \le s_1, s_2 \le n+m,\ s_1 \ne s_2;\ u_{s_1} = 1,\ u_{s_2} = -1,\ \theta_{s_2} > 0,\ \forall i \notin \mathbf{s}\ u_i = 0\}$.
2) $I_2 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2),\ 1 \le s_1, s_2 \le m,\ s_1 \ne s_2,\ y_{s_1} = y_{s_2};\ u_{s_1} = 1,\ \theta_{s_1} < \rho C^* \pi_{s_1},\ u_{s_2} = -1,\ \theta_{s_2} > 0,\ \forall i \notin \mathbf{s}\ u_i = 0\}$.
3) $I_3 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2),\ m+1 \le s_1, s_2 \le n,\ s_1 \ne s_2,\ y_{s_1} = y_{s_2};\ u_{s_1} = 1,\ \theta_{s_1} < C\pi_{s_1},\ u_{s_2} = -1,\ \theta_{s_2} > 0,\ \forall i \notin \mathbf{s}\ u_i = 0\}$.
4) $I_4 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2),\ m+1 \le s_1, s_2 \le n,\ s_1 \ne s_2,\ y_{s_1} \ne y_{s_2},\ \forall i \notin \mathbf{s}\ u_i = 0;\ u_{s_1} = u_{s_2} = 1,\ \theta_{s_1} < C\pi_{s_1},\ \theta_{s_2} < C\pi_{s_2}$ or $u_{s_1} = u_{s_2} = -1,\ \theta_{s_1} > 0,\ \theta_{s_2} > 0\}$.
5) $I_5 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2, s_3),\ 1 \le s_1, s_2 \le m,\ n+1 \le s_3 \le n+m,\ s_1 \ne s_2,\ y_{s_1} \ne y_{s_2},\ \forall i \notin \mathbf{s}\ u_i = 0;\ u_{s_1} = u_{s_2} = 1,\ \theta_{s_1} < \rho C^* \pi_{s_1},\ \theta_{s_2} < \rho C^* \pi_{s_2},\ u_{s_3} = -2,\ \theta_{s_3} > 0$ or $u_{s_1} = u_{s_2} = -1,\ \theta_{s_1} > 0,\ \theta_{s_2} > 0,\ u_{s_3} = 2\}$.
6) $I_6 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2, s_3),\ 1 \le s_1 \le m,\ m+1 \le s_2 \le n,\ n+1 \le s_3 \le n+m,\ y_{s_1} = y_{s_2},\ \forall i \notin \mathbf{s}\ u_i = 0;\ u_{s_1} = 1,\ \theta_{s_1} < \rho C^* \pi_{s_1},\ u_{s_2} = -1,\ \theta_{s_2} > 0,\ u_{s_3} = -1,\ \theta_{s_3} > 0$ or $u_{s_1} = -1,\ \theta_{s_1} > 0,\ u_{s_2} = 1,\ \theta_{s_2} < C\pi_{s_2},\ u_{s_3} = 1\}$.
7) $I_7 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2, s_3),\ 1 \le s_1 \le m,\ m+1 \le s_2 \le n,\ n+1 \le s_3 \le n+m,\ y_{s_1} \ne y_{s_2},\ \forall i \notin \mathbf{s} u_i = 0;\ u_{s_1} = u_{s_2} = 1,\ \theta_{s_1} < \rho C^* \pi_{s_1},\ \theta_{s_2} < C\pi_{s_2},\ u_{s_3} = -1,\ \theta_{s_3} > 0$ or $u_{s_1} = u_{s_2} = -1,\ \theta_{s_1} > 0,\ \theta_{s_2} > 0,\ u_{s_3} = 1\}$.
8) $I_8 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2, s_3),\ 1 \le s_1, s_2 \le m,\ m+1 \le s_3 \le n,\ s_1 \ne s_2,\ y_{s_1} \ne y_{s_2},\ y_{s_3} = y_{s_2},\ \forall i \notin \mathbf{s}\ u_i = 0;\ u_{s_1} = 1,\ \theta_{s_1} < \rho C^* \pi_{s_1},\ u_{s_2} = -1,\ \theta_{s_2} > 0,\ u_{s_3} = 2,\ \theta_{s_3} < C\pi_{s_3}$ or $u_{s_1} = -1,\ \theta_{s_1} > 0,\ u_{s_2} = 1,\ \theta_{s_2} < \rho C^* \pi_{s_2},\ u_{s_3} = -2,\ \theta_{s_3} > 0\}$.
9) $I_9 \triangleq \{\mathbf{u_s} | \mathbf{s} = (s_1, s_2, s_3),\ 1 \le s_1, s_2 \le m,\ m+1 \le s_3 \le n,\ s_1 \ne s_2,\ y_{s_1} \ne y_{s_2},\ y_{s_3} = y_{s_1},\ \forall i \notin \mathbf{s}\ u_i = 0;\ u_{s_1} = 1,\ \theta_{s_1} < \rho C^* \pi_{s_1},\ u_{s_2} = -1,\ \theta_{s_2} > 0,\ u_{s_3} = -2,\ \theta_{s_3} > 0$ or $u_{s_1} = -1,\ \theta_{s_1} > 0,\ u_{s_2} = 1,\ \theta_{s_2} < \rho C^* \pi_{s_2},\ u_{s_3} = 2,\ \theta_{s_3} < C\pi_{s_3}\}$.

It can be proved that moving from any feasible point $\boldsymbol{\theta}^{\text{old}}$ in the direction $\mathbf{u_s} \in \cup I_i$ still satisfies constraints (4,5). The optimization step is $\boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^{\text{old}} + \lambda^*(\mathbf{s})\mathbf{u_s}$, where $\mathbf{u_s} \in \cup I_i$ and the step size $\lambda^*(\mathbf{s})$ maximizes the corresponding cost function, $\psi(\lambda') = D(\boldsymbol{\theta}^{\text{old}} + \lambda' \mathbf{u_s})$, while satisfying the constraints. Let $\mathbf{g}(\boldsymbol{\theta}^{\text{old}})$ and H respectively be the gradient at point $\boldsymbol{\theta}^{\text{old}}$ and the Hessian of cost function (3). Then the Taylor expansion of $\psi(\lambda')$ at point $\lambda' = 0$ yields

$$\lambda\left(\boldsymbol{\theta}^{\text{old}}, \mathbf{s}\right) = \arg\max_{\lambda' \ge 0} \psi\left(\lambda'\right) = -\frac{\mathbf{g}\left(\boldsymbol{\theta}^{\text{old}}\right)^T \mathbf{u_s}}{\mathbf{u_s^T} H \mathbf{u_s}} \tag{9}$$

which can be used to find the optimal feasible direction

$$\begin{aligned}\mathbf{s}^{(i)} &= \arg \max_{\substack{\mathbf{t}:\mathbf{t}=\left(s_1^{(i)}, s_2^{(i)}\right) \\ \mathbf{u_t} \in I_i}} D\left(\boldsymbol{\theta}^{\text{old}} + \lambda'\left(\boldsymbol{\theta}^{\text{old}}, \mathbf{t}\right)\mathbf{u_t}\right) - D\left(\boldsymbol{\theta}^{\text{old}}\right) \\ &= \arg \max_{\substack{\mathbf{t}:\mathbf{t}=\left(s_1^{(i)}, s_1^{(i)}\right) \\ \mathbf{u_t} \in I_i}} \frac{-\left(\mathbf{g}\left(\boldsymbol{\theta}^{\text{old}}\right)^T \mathbf{u_t}\right)^2}{\mathbf{u_t^T} H \mathbf{u_t}}\end{aligned} \tag{10}$$

Now that the best direction is chosen, we need to ensure that $\lambda(\boldsymbol{\theta}^{\text{old}}, \mathbf{s})$ does not overstep the bounding constraints (6,7). Therefore, we define the following clipping function:

$$\lambda^*\left(\boldsymbol{\theta}^{\text{old}}, \mathbf{s}^{(i)}\right) = \min_{\substack{j,k \in \mathbf{s}^{(i)} \\ u_j < 0,\, u_k > 0}} \left\{\lambda\left(\boldsymbol{\theta}^{\text{old}}, \mathbf{s}^{(i)}\right), \frac{\Delta_k - \theta_k^{\text{old}}}{u_k}, \left|\frac{\theta_j^{\text{old}}}{u_j}\right|\right\}$$

where $\Delta_k = \rho C^* \pi_k$ for $1 \le k \le m$, and $\Delta_k = C\pi_k$ for $m+1 \le k \le n$. By iteratively choosing the best direction and applying the clipping function to the step size, the algorithm converges while satisfying all the constraints. The schematic of the LULUPAPI iterative optimizer is depicted in figure 1.

IV. EXPERIMENT

The ARDS dataset used in this study consisted of 485 patients with moderate hypoxia or acute hypoxic respiratory failure, recorded at the University of Michigan Hospital.

TABLE I

CLASSIFICATION RESULTS ON ARDS DATASET USING TWO SVM AND FOUR LULUPAPI CASES: (1) NO UNCERTAINTY, I.E. $\pi_i = 1$ FOR $1 \leq i \leq n$ (LEARNING USING PARTIALLY AVAILABLE PRIVILEGED INFORMATION), (2) UNCERTAINTY ONLY FOR NON-PRIVILEGED INFORMATION I.E. $\pi_i = 1$ FOR $1 \leq i \leq m$, (3) UNCERTAINTY ONLY FOR PRIVILEGED INFORMATION I.E. $\pi_i = 1$ FOR $m+1 \leq i \leq n$ AND (4) UNCERTAINTY FOR ALL SAMPLES.

	Train				Test			
	Accuracy	Sensitivity	Specificity	AUC	Accuracy	Sensitivity	Specificity	AUC
SVM	88.61	80.43	91.76	86.10	89.27	76.58	90.33	83.46
SVM with Uncertainty	87.56	78.74	90.96	84.85	89.52	80.03	90.32	85.17
LULUPAPI (1)	88.09	81.28	90.72	86.00	88.78	82.23	89.34	85.78
LULUPAPI (2)	88.52	84.21	90.19	87.20	88.28	83.47	88.69	86.08
LULUPAPI (3)	87.53	83.28	89.18	86.23	87.99	85.26	88.22	86.74
LULUPAPI (4)	87.96	83.59	89.65	86.62	88.38	85.40	88.63	**87.01**

The non-privileged information included 25 clinical features (such as temperature, heart rate, etc.) at two-hour intervals from the patient's EHR record. Each patient in the dataset was reviewed by 2 or 3 clinical experts for the diagnosis of ARDS, as well as the time of ARDS onset. In order to account for their level of confidence, clinicians reported their diagnosis confidence, denoted by l_i, using a 1-8 scale in which 1 is no ARDS with high confidence and 8 is ARDS with high confidence. To quantitatively measure uncertainty in the labels (the π_i in equation 3), we used a margin weight generator $\pi_i = (|l_i - p_1| - p_2)p_3 + p_4$. We set $p_1 = 4.5$, $p_2 = 3$, $p_3 = 0.2$, and $p_1 = 0.9$. This scaled the l_i with range 1-8 into the range 0.4-1 such that high-confidence cases l_i=1 and l_i=8 were mapped to $\pi_i = 1$ and low-confidence cases l_i=4 and l_i=5 were mapped to $\pi_i = 0.4$. The privileged information was a one-dimensional feature representing the average of chest x-ray evaluation scores, given by three clinicians, such that 8 designated high confidence ARDS and 1 high confidence non-ARDS. In order to avoid bias toward patients, the dataset was split 2/3 and 1/3 into training and holdout sets such that all samples from the same patient were kept exclusively in either training or testing. This yielded 323 patients in the training data, and the rest in the holdout set. Due to the strong inter-dependency between samples of longitudinal patient data, the IID assumption was not valid. Hence the time-series sampling method proposed in [18] was performed to reduce the inter-correlation among the longitudinal clinical data from each patient used in model training and thereby limit overfitting. After sampling, there were 4661 samples in the training data, of which 4317 had privileged information, across 1298 ARDS cases. Since there was no sampling in the test dataset, there were 9362 samples in the test dataset. Within the training set, 5-fold cross-validation was performed for 4-dimensional hyperparameter optimization in the interval $[0.1, 5]$ for C and C^*, $[1, 5]$ for ρ, and $[0.5, 2]$ for γ. The bins were split by patient. At each round, four bins were used for training and one for testing with linear kernels. Table I summarizes the results of SVM, SVM with label uncertainty, and four cases of LULUPAPI. As can be seen, while separate incorporation of label uncertainty and privileged information improved the SVM result (SVM with Uncertainty and LULUPAPI (1) in Table I), the best test AUC occurred by simultaneous incorporation of label uncertainty and privileged information (LULUPAPI (4) in Table I).

REFERENCES

[1] G. Rubenfeld, E. Caldwell, E. Peabody, J. Weaver, D. Martin, E. Stern, and L. Hudson, "Incidence and outcomes of acute lung injury," *The New England journal of medicine*, vol. 353, no. 16, pp. 1685–1693, 2005.

[2] "Ventilation with lover tidal volumes as compared with traditional tidal volumes for acute lung injury and the acute respiratory distress syndrome. the acute respiratory distress syndrome network," *The New England journal of medicine*, vol. 342, no. 18, pp. 1301–1308, 2000.

[3] G. Bellani, J. Laffey, T. Pham, E. Fan, L. Brochard, L. Esteban, A. Gattinoni, F. van Harn, A. Larsson, D. McAuley, M. Ranieri, G. Rubenfeld, B. Thompson, H. Wrigge, A. Slutsky, and A. Pesenti, "Epidemiology, patterns of care, and mortality for patients with acute respiratory distress syndrome in intensive care unites in 50 countries," *JAMA: the journal of the American Medical Association*, vol. 315, no. 8, pp. 788–800, 2016.

[4] B. Frenay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2014.

[5] N. Natarajan, I. Dhillon, P. Ravikuman, and A. Tewari, "Learning with noisy labels," *Advances in neural information processing systems*, 2013.

[6] Y. Duan and O. Wu, "Learning with auxiliary less-noisy labels," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–6, 2016.

[7] S. Vembu and S. Zilles, "Interactive learning from multiple noisy labels," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2016.

[8] V. Vapnik and R. Izmailov, "Learning using privileged information: Similarity control and knowledge transfer," *Journal of Machine Learning Research*, vol. 16, pp. 2023–2049, 2015.

[9] V. Sharmanska, N. Quadrianto, and C. Lampert, "Learning to rank using privileged information," *The IEEE International Conference on Computer Vision*, pp. 825–832, 2013.

[10] B. Ribeiro, C. Silva, N. Chen, A. Vieira, and J. Carvalho das Neves, "Enhanced default risk models with svm+," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10 140–10 152, 2012.

[11] L. Liang and V. Cherkassky, "Connection between svm+ and multi-task learning," *Connection between SVM+ and multi-task learning*, pp. 2048–2054, 2008.

[12] V. Vapnik, *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

[13] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural networks*, vol. 22, no. 5-6, pp. 544–557, 2009.

[14] V. Vapnik and R. Izmailov, "Learning using privileged information: similarity control and knowledge transfer." *Journal of machine learning research*, vol. 16, no. 20232049, p. 55, 2015.

[15] D. Pechyony, R. Izmailov, A. Vashist, and V. Vapnik, "Smo-style algorithms for learning using privileged information." in *DMIN*, 2010, pp. 235–241.

[16] D. Pechyony and V. Vapnik, "Fast optimization algorithms for solving svm+," *Stat. Learning and Data Science*, vol. 1, 2011.

[17] J. C. Platt, "12 fast training of support vector machines using sequential minimal optimization," *Advances in kernel methods*, pp. 185–208, 1999.

[18] N. Reamaroon, M. W. Sjoding, K. Lin, T. J. Iwashyna, and K. Najarian, "Accounting for label uncertainty in machine learning for detection of acute respiratory distress syndrome," *IEEE Journal of Biomedical and Health Informatics*, 2018.