Learning from Positive and Unlabeled Multi-Instance Bags in Anomaly Detection

Lorenzo Perini
lorenzo.perini@kuleuven.be
KU Leuven, Dept. of Computer Science; Leuven.AI,
B-3000 Leuven, Belgium

Vincent Vercruyssen
vincent.vercruyssen@kuleuven.be
KU Leuven, Dept. of Computer Science; Leuven.AI,
B-3000 Leuven, Belgium

Jesse Davis
jesse.davis@kuleuven.be
KU Leuven, Dept. of Computer Science; Leuven.AI,
B-3000 Leuven, Belgium

ABSTRACT

In the multi-instance learning (MIL) setting instances are grouped together into bags. Labels are provided only for the bags and not on the level of individual instances. A positive bag label means that at least one instance inside the bag is positive, while a negative bag label restricts all the instances in the bag to be negative. MIL data naturally arises in many contexts, such as anomaly detection, where labels are rare and costly, and one often ends up annotating the label for sets of instances. Moreover, in many real-world anomaly detection problems, only positive labels are collected because they usually represent critical events. Such a setting, where only positive labels are provided along with unlabeled data, is called Positive and Unlabeled (PU) learning. Despite being useful for several use cases, there is no work dedicated to learning from positive and unlabeled data in a multi-instance setting for anomaly detection. Therefore, we propose the first method that learns from PU bags in anomaly detection. Our method uses an autoencoder as an underlying anomaly detector. We alter the autoencoder's objective function and propose a new loss that allows it to learn from positive and unlabeled bags of instances. We theoretically analyze this method. Experimentally, we evaluate our method on 30 datasets and show that it performs better than multiple baselines adapted to work in our setting.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

Multi-Instance Learning, PU Learning, Anomaly Detection

ACM Reference Format:
Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. 2023. Learning from Positive and Unlabeled Multi-Instance Bags in Anomaly Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599409

1 INTRODUCTION

Multi-instance learning (MIL) [10, 24, 48] is a form of weakly supervised learning where the learner has access to sets of instances, called bags. Most importantly, labels are provided on the bag level and not for each individual instance. In MIL for binary classification, a positive bag contains at least one instance that belongs to the positive class, but it is not known which ones are responsible for the bag label. In contrast, a negative bag only contains instances that belong to the negative class. Multi-instance data naturally arises in many applications like drug activity prediction [10], video [29], document [49], and sound [5] classification.

MIL data also naturally arises in anomaly detection [7, 16] where the task is to detect unexpected instances [13, 33, 34]. Two illustrative examples are monitoring resource usage in a retail store to avoid water leaks [40] and collecting sensor data from wind turbines to detect blade icing [46]. In both cases, experts tend to provide coarse-grained labels by flagging an anomaly on the level of a day (or longer), while the data is collected on a more fine-grained level. However, the anomalous behavior is not necessarily present for the majority of the flagged time period. Moreover, experts need the anomaly detection system not only to flag anomalous behaviors on a daily level but rather at the exact time the anomalous behavior occurs, as it is practically relevant for decision-making (e.g., a spike in water usage at night requires different response than one during opening hours of the store). As a result, constructing instances at a coarse granularity by defining features over the full window has two evident risks: (i) making the instance seem more normal than it is when aggregating over both normal and abnormal behaviors; (ii) losing the connection between the observed behavior and the time it occurred. This requires posing the problem as a MIL task.

Most anomaly detection scenarios are naturally characterized by small amounts of labeled data and lots of unlabeled data [26, 39] because anomalies are rare and hard to acquire. Moreover, one ends up only annotating positive (i.e., anomalous) bags because anomalies are usually connected to critical events. Unfortunately, it is not safe to assume that all unlabeled instances are non-anomalous because some failures may not be detected by the experts. Hence, we are dealing with a PU problem [4, 12], which is a special case of binary classification where the learner has only access to positive (P) labels and unlabeled (U) data. Alas, there is no research on anomaly detection methods that handle both the MIL and PU settings.

In this paper, we try to fill this gap by investigating the combination of MIL and PU learning for anomaly detection and propose a novel algorithm called **PUMA** (Positive and Unlabeled Multi-instance Anomaly detector) for this setting. Our approach adopts the standard PU learning assumption that the observed positive

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599409

labels are "Selected Completely At Random" (SCAR) [4, 12] on the bag level for MIL. This assumption means that we have the same constant probability of observing every positive bag label. Additionally, it assumes that there are fewer positive instances (i.e., anomalies) than negative instances, which is standard in anomaly detection. PUMA uses an autoencoder as the base anomaly detector. We design a new loss function for training the autoencoder that extends the typical unlabeled objective of minimizing the reconstruction error with a second labeled component that exploits the bag labels. Conceptually, this loss uses the unlabeled bags to force the model to learn what constitutes the normal instance behavior and the positive bag labels to force it to learn to discriminate between anomalous and normal behaviors.

Overall, we make **four contributions**. First, we introduce the problem of learning from PU bags in anomaly detection, which has several use cases. Second, we develop PUMA that can learn a model capable of assigning both instance and bag labels. Third, we theoretically analyze PUMA's ability to learn from large bags. Finally, we experimentally evaluate PUMA on 9 real-world datasets, divided into two use cases, and 21 benchmark datasets.

2 PRELIMINARIES AND RELATED WORK

Multi-Instance Learning. Multi-instance learning (MIL) [10, 24, 48] refers to the setting where a learner has access to instances but labels are given only on a bag level, where a bag is a set of instances. Let $X = \{x_1, \dots, x_N\}$ be a dataset with N instances such that $x_i \in \mathbb{R}^d$ for $i \leq N$. Formally, we assume that the instances are provided within M bags $B = \{x_1, \dots, x_k\}$, where k is the bag size such that $k \times M = N$. We indicate by Y_B the label of the bag B, and by y_x the label for the instance x. A positive bag label $Y_B = 1$ indicates that at least one instance in the bag is positive (i.e., there exists $i \leq k$ such that $y_{x_i} = 1$), while a negative bag label $Y_B = 0$ means that all the instances are negative (i.e., for all $i \leq k$ we have $y_{x_i} = 0$). When applied to anomaly detection, the task of MIL is to assign labels either to the bags [19], or to the instances [23] by introducing a probability function [17, 18, 34]. In contrast, we build an anomaly detector that performs **both**: we develop a bag and an instance probability function to assign labels to instances and bags.

Positive and Unlabeled Learning. In Positive and Unlabeled (PU) learning [4] the model only has access to positive labeled instances and unlabeled instances. In this setting, we assume that only *positive bag labels* are provided along with unlabeled bags. This means that $Y_B = 1$ for $B \in P$, while Y_B is unknown for $B \in U$, where P and U are, respectively, the set of labeled and unlabeled bags. Because we assume that labels are SCAR, each (positive) bag has constant probability c (i.e., the label frequency) to be labeled. Unfortunately, there is only limited literature about MIL with PU bags [2, 11, 38, 42]. Other literature focuses on semi-supervised MIL [14, 41, 43, 47], where, commonly, models are based on a loss function that separately learns from labeled and unlabeled bags.

Anomaly detection. Anomaly detection algorithms assign a score to each instance in a dataset representing its degree of anomalousness [25, 28]. This paper uses an autoencoder [1] as anomaly detector. An autoencoder (AE) is a neural network composed of an encoder that maps the input instance x into a lower-dimensional hidden space, and a decoder that maps it back to the original space. The goal is to find a lower dimensional representation that still enables accurate reconstruction of the input. In anomaly detection, a common way to assign anomaly scores through an AE is to use its reconstruction error: $a_\theta(x) = \|x - \text{AE}_\theta(x)\|^2$, where $\| \cdot \|$ is the Euclidean norm, $\text{AE}_\theta(x)$ is the reconstructed input by the autoencoder with θ as parameters (i.e., the network weights). Because the AE is trained to minimize $a_\theta(x)$ on the training set, the higher the reconstruction error for a test instance, the more anomalous it is. In this work, we refer to *the anomaly class as the positive one*. Similarly to our work, Iwata et al. [19] introduce the inexact AUC and use the autoencoder to learn from bag labels. However, they assume a fully labeled setting and that positive bags contain exactly one anomaly.

3 A PU MULTI-INSTANCE ANOMALY DETECTION FRAMEWORK

This paper tackles the following problem:

Given: a set P of positive (anomalous) labeled bags such that $Y_B = 1$ if $B \in P$, a set U of unlabeled bags such that Y_B is unknown for $B \in U$;

Do: find a bag probability function F and an instance probability function f that assign, respectively, bag and instance probabilities of being anomalous.

Learning from PU bags in anomaly detection has four challenges. First, semi-supervised anomaly detectors assume that labels are associated with the instances, while we are provided with (only positive) labels on a bag level. Propagating such labels to an instance level is challenging. Second, we lack negatively labeled bags which may affect the model's training as it is naturally inclined to overfit toward the positive class, if not properly corrected. Third, some anomalies may not follow specific patterns. Thus, anomalous instances in the positively labeled bags may not be fully representative of the positive instance class. Fourth, real-world applications need predictions both on a bag (F) and on an instance (f) level. However, most of the existing methods focus only on few distinguishable positive instances that trigger the bag label (e.g., using the max of a scoring function) and treat all the remaining instances as negative [23, 36]. This erodes the performance on the instance level, as bags can contain multiple positive instances.

In this paper, we introduce **PUMA** (Positive and Unlabeled Multi-instance Anomaly detector), a novel loss-based approach that learns from positive and unlabeled bags in an anomaly detection setting and assigns positive (anomalous) class probabilities on both an instance and a bag level. Specifically, PUMA trains an Autoencoder with a two-component loss function $L = L_u + L_p$. Through the first component (L_u), it uses the unlabeled bags to model the distribution of the seen examples in order to detect anomalies that do not follow any patterns (e.g., novelties) at the test time (Section 3.1). Through the second component (L_p), it exploits the positive bag labels to learn to discriminate between positives and negatives (Section 3.2). Finally, L can be minimized using the common back-propagation technique (Section 3.3).

3.1 The unlabeled loss component L_u

Because anomaly detection problems usually have few (or no) labels, anomaly detectors need to leverage a large amount of unlabeled data. For this task, we select the autoencoder for two reasons. First, it is accurate at detecting novel examples [9], which makes the model able to capture anomalies that do not follow any pattern. Second, it is robust to slightly contaminated data (i.e., the training set contains unlabeled anomalies), particularly when its structure is limited to few layers and neurons [35].

Because the input is a set of bags, we derive the bag-wise reconstruction error $\mathcal{R}_\theta(B) = \frac{1}{k}\sum_{j=1}^{k} a_\theta(x_j)$ for an unlabeled bag $B = \{x_1, \ldots, x_k\}$ as the average of the instance reconstruction error $a_\theta(x_j)$ (for $j \leq k$). The unlabeled component L_u is computed as the average over all the unlabeled bags:

$$L_u := \frac{1}{|U|}\sum_{B \in U} \mathcal{R}_\theta(B) = \frac{1}{k|U|}\sum_{B \in U}\sum_{x_j \in B} a_\theta(x_j). \tag{1}$$

Varying the autoencoder's parameters θ has an impact on both f (instance function) and F (bag function), as shown in Sec. 3.2.

3.2 The labeled loss component L_p

Building a loss function that uses positive bag labels requires setting up a learning scheme where: (1) we parametrize the instance probability function f, (2) we link it to the bag probability function F, (3) we collect negative bag labels, and (4) we use the bag labels to measure how accurate F is. Each of these steps has hidden challenges that we tackle as follows.

1. Mapping instance scores to probabilities. Because the Autoencoder's instance anomaly scores are in $[0, +\infty)$, there is no guarantee of comparable scaling across different bags [27]. Thus, we transform the instance reconstruction errors $a_\theta(x)$ into calibrated probabilities by applying Platt scaling [15, 30]:

$$f(x) = \widehat{\mathbb{P}(y_x = 1)} = \frac{1}{1 + \exp(-\alpha a_\theta(x) - \beta)} \tag{2}$$

where f depends on the autoencoder's parameters θ, and on the two new calibration parameters $\alpha, \beta \in \mathbb{R}$. Note that the principle itself is not restricted to this particular choice of f, but one could apply any transformation to $[0, 1]$. We choose Platt scaling because it is widely used in the literature.

2. Transforming instance probabilities into bag probabilities. Computing the bag probability by taking the max instance probability means that the model is only updated based on the single instance that triggers the bag label [19, 23], and hence ignores the information present in the other examples. On the other hand, taking an unweighted average of the instance probabilities would make a bag that contains a small number of anomalies seem more normal than it is. One solution would be to use the Noisy-OR approach [24], which computes the bag probability of being positive as "one minus the probability that all the instances are negatives". However, we show in Section 4 that the Noisy-OR is ill-conditioned when the bag size k is large. Therefore, we propose a **weighted Noisy-OR** approach, instead. Our key insight is that *the instances with the highest and lowest positive probabilities should have higher weights because they contribute more to defining the bag label*:

$$F(B) = \widehat{\mathbb{P}(Y_B = 1)} = 1 - \prod_{x_j \in B}\left(1 - f(x_j)\right)^{w_j}, \tag{3}$$

where F depends on θ, α and β through f, and w_j is the weight for x_j. Note that if we used $1 - f(x_j)^{w_j}$ instead of $(1 - f(x_j))^{w_j}$, we would modify the instance probabilities $f(x_j)$ and aggregate them with equal weight, which is not consistent with our insight.

The choice of the weights should reflect that a bag probability $F(B)$ is mainly decided by the highest and lowest instance probabilities $f(x_j)$ in the bag. Thus, we (1) rank the instances in each bag according to their probabilities to define the local "highest/lowest" probabilities, (2) assign a score to each rank that measures the instance contribution for the bag label, and (3) transform such score into a proper weight w_j for Eq. 3. Éach step works as follows:

First, we rank the positive instance probabilities in ascending order using a ranking map $\rho_f \colon \mathbb{R}^k \to \{0, \ldots, k-1\}$, which assigns the instance x_j to its rank r

$$\rho_f(x_j) = r \in \{0, \ldots, k-1\} \Longleftrightarrow \begin{cases} |\{x \in \{x_1, \ldots, x_k\} \colon f(x) < r| = r \\ |\{x \in \{x_1, \ldots, x_k\} \colon f(x) > r| = k - r - 1 \end{cases}$$

Then, we normalize the rankings to $[0, 1]$ by dividing them by $k - 1$.

Second, we introduce a weighting function $S \colon [0, 1] \to \mathbb{R}$ that gives high weights to both high and low rankings. Our reasoning is that high-ranking instances should indicate how "positive" the bag label is, while low-ranking instances (expected to be normal) should indicate the opposite. For the sake of simplicity, we imagine such a function to have two peaks (on 0 for low rankings and 1 for high rankings) and be flat (almost null) in between. A natural choice of S is $\mathcal{N}_0 + \mathcal{N}_1$ (restricted to $[0, 1]$), where $\mathcal{N}_a$ is the Gaussian density function with mean a and standard deviation 0.1:

$$S\left(\frac{\rho_f(x_j)}{k-1}\right) = \mathcal{N}_0\left(\frac{\rho_f(x_j)}{k-1}\right) + \mathcal{N}_1\left(\frac{\rho_f(x_j)}{k-1}\right).$$

Note that the principle itself is not restricted to this particular choice of functional form for S. One could apply a different map, but the detailed theoretical results in Sec. 4 would naturally be different.

Third, we apply S to each instance's ranking in the bag and normalize such values as

$$w_j = S\left(\frac{\rho_f(x_j)}{k-1}\right) \Bigg/ \sum_{q \leq k} S\left(\frac{\rho_f(x_q)}{k-1}\right).$$

We assign the weight w_j to the instance x_j of each bag and use it to derive the bag probability. Note that alternative methods like [21] cannot be used in this setting because we exclude the possibility that negative bags have positive instances.

3. Selecting the R ***reliable negatives.*** Learning from only positive bag labels would allow the model to overfit towards the positive class. Therefore, we propose to select $|R| = |P|$ negative bags in order to transform the problem into a *classification task with balanced classes.* PUMA selects the $|R|$ bags among B with the lowest positive probability $F(B)$ as the reliable negatives. This requires the assumption that the positive and negative classes are separable to be true [3, 8], which is likely to be the case in anomaly detection because anomalies are usually well-separated from normal examples. However, this selection rule may introduce bias because the negative labels are not selected in an i.i.d. way. We attempt to limit the amount of bias by having the model re-select different negatives R in each iteration (i.e., epoch) of the training loop. In this way, PUMA progressively refines its learned definition of the negative class. In Section 5 we empirically analyze the value $|R|$ and show that setting it to $|P|$ to keep the classes balanced yields good results.

4. Learning f and F from PU bags. Evaluating the bag probability function F requires comparing its output with the bag labels. We use the log-likelihood of the positive and (self-generated) negative bag labels under the corresponding bag probabilities produced by F, and take its negative value as loss function L_p. Formally, assuming that labels follow a Bernoulli distribution with parameter $F(B)$, we build the loss function L_p as

$$L_p = -\log\left(\prod_{B\in P} F(B) \prod_{B\in R}(1-F(B))\right) + \lambda\left(\alpha^2+\beta^2\right) \tag{4}$$

where $\lambda(\alpha^2+\beta^2)$ avoids overfitting. We set λ to 0.01.

We derive PUMA's loss function as $L = L_u + L_p$ (Eq. 1 and 4) and find the optimal parameters θ, α, β by minimizing L. This allows the model to (a) learn the distribution of the normal instances and (b) discriminate between positives and negatives.

Note that we do not scale the two components L_u (weight as 1 bag) and L_p (weight as $|P|$+$|R|$ bags) for two reasons. First, positive labels are rare and we want to exploit them as much as possible. Second, the more positive labels the more likely they are representative of the whole positive class, and the less we need to detect novelties (via the unlabeled component). This is reasonable as several use cases have a limited number of different novelties that can occur.

3.3 Training with back-propagation.

Because back-propagation requires computing the gradient of L, we show that L is differentiable by proving that both L_u and L_p are differentiable.

1)L_u. The unlabeled component L_u (Eq. 1) depends only on the Euclidean norm in $a_\theta(x)$, which is differentiable on θ by definition.

2)L_p. In the label component, the penalization term is obviously differentiable. The log-likelihood is differentiable if and only if $F = F_{\theta,\alpha,\beta}$ is. Note that $F(B) = 1 - \prod_{j\le k}(1-(1-f(x_j))^{w_j})$ depends on the parameters θ, α, β through $f = f_{\theta,\alpha,\beta}$ and w_j. Because f is a sigmoid function on top of the reconstruction error, it is differentiable. In addition, f cannot take the extreme values 0 and 1 because it is a sigmoid function, which means that $(1-f(x))^{S(x)}$ is differentiable for any x if and only if the function S is. Since Platt scaling is Lipschitz-continuous, we reasonably assume that small variations in the parameters lead to small variations in the instance probabilities f but do not change the instance ranking. Therefore, S is differentiable with null gradients. Note that this does not imply that the whole gradient is null because multiple terms appear due to the chain rule of derivation.

4 THEORETICAL ANALYSIS

The standard noisy-OR approach is widely used in the MIL literature [24, 44]. In this section we answer the question: *why is the weighted noisy-OR necessary for learning, as opposed to the standard noisy-OR?* We do so in two steps. First, we illustrate the noisy-OR's drawback that does not allow PUMA to learn for large bag sizes. Second, we show that the weighted noisy-OR does not present the same issue.

1. Standard noisy-OR drawback. Given a bag B, the standard noisy-OR derives the probability that B is positive as "one minus the probability that all the instances are negatives"

$$\text{(Noisy-OR)} \qquad \widehat{\mathbb{P}(Y_B=1)} = 1 - \prod_{j\le k}(1-f(x_j)).$$

However, for large values of k, this probability converges to 1 regardless of the instance probabilities $f(x_j)$:

Theorem 4.1. *Given an instance probability function $f\colon \mathcal{X}_I \longrightarrow (0,1)$ and a bag $B = \{x_1,\dots,x_k\}$, the standard noisy-OR approach produces bag probabilities of being positive that converge exponentially to* 1 *for $k \longrightarrow +\infty$.*

Proof. Because f maps instances to $(0,1)$, the standard noisy-OR approach assigns bag probabilities always strictly lower than 1 and greater than 0, i.e., $0 < f(x_j) = \widehat{\mathbb{P}(y_x=1)} < 1$. Assume without loss of generality, that $f(x_j)$ are i.i.d. random variables distributed in $[0,1]$ such that 0 and 1 are events with 0 probability. Therefore, $0 < \mathbb{E}[f(x_j)] < 1$. Thus, increasing the bag size k (i.e., for $k \longrightarrow +\infty$) the expected bag probability $\mathbb{E}[\widehat{\mathbb{P}(Y_B=1)}]$ is equal to

$$\mathbb{E}\left[1-\prod_{j\le k}(1-f(x_j))\right] = 1-\prod_{j\le k}\mathbb{E}\left[(1-f(x_j))\right] = 1-\left(1-\mathbb{E}\left[f(x_j)\right]\right)^k$$

and always goes exponentially to 1, as the product of values strictly lower than 1 decreases monotonically with respect to k. □

As a result, in our setting the standard noisy-OR would constantly output bag probabilities equal to 1 for large values of k. For example, if the instance probabilities were distributed uniformly in $[0,1]$, then the expected bag probability would be $1-\frac{1}{2^k}$. With $k=30$, it would be $1-1\times10^{-10}$, which is likely to be dominated by computational machine errors. In contrast, the weighted noisy-OR does not present the same problem.

2. The weighted noisy-OR converges to $(0,1)$**.** Because the highest and lowest instance probabilities have the largest effect on estimated bag probability, the weighted noisy-OR is not affected by large values of k. We first provide an intermediate result:

Theorem 4.2. *Given a bag $B = \{x_1,\dots,x_k\}$, the weights normalization constant can be approximated as $\sum_{j\le k} w_j \approx 2k\Phi_0\left(\frac{1}{2k}\right)$ for large values of k, where Φ_0 is the cumulative of $\mathcal{N}_0$.*

Proof. Given k instances, the normalized ranking of their probabilities is an equally spaced grid of $[0,1]$ (extremes included) with gap equal to $\frac{1}{k}$. Thus, summing all the weights w_q for $q\le k$ and multiplying them by $\frac{1}{k}$ is a fair approximation of the area below the density $\mathcal{N}_0(t)+\mathcal{N}_1(t)$ for $t\in\left[-\frac{1}{2k}, 1+\frac{1}{2k}\right]$. Note that we need to slightly extend the $[0,1]$ interval when computing the area in order to include the additional term due to the discretization of the area. Indicating by $\Phi_0(t)$ and $\Phi_1(t)$ the cumulative distributions in

t of, respectively, $\mathcal{N}_0$ and $\mathcal{N}_1$, it follows that

$$\begin{aligned}\frac{1}{k}\sum_{j\leq k} w_j &\approx \Phi_1\left(1+\frac{1}{2k}\right)-\Phi_1\left(-\frac{1}{2k}\right)+\Phi_0\left(1+\frac{1}{2k}\right)-\Phi_0\left(-\frac{1}{2k}\right)\\ &= 2\left[\Phi_0\left(1+\frac{1}{2k}\right)-\Phi_0\left(-\frac{1}{2k}\right)\right] = 2\left[0.5+\Phi_0\left(\frac{1}{2k}\right)-\Phi_0\left(0\right)\right]\\ &= 2\cdot\Phi_0\left(\frac{1}{2k}\right) \implies \sum_{j\leq k} w_j \approx 2k\cdot\Phi_0\left(\frac{1}{2k}\right),\end{aligned}$$

where the equality on the first line comes from the symmetry of the two normal random variables, while the second line steps depend on the properties of the Gaussian random variables. □

Thanks to this result, we conclude that the weighted noisy-OR converges to a value that can be neither 0 nor 1:

Theorem 4.3. *For a bag B, the probability $\widehat{\mathbb{P}(Y_B=1)}$ converges in $(0,1)$ for $k \longrightarrow +\infty$.*

Proof. By using the Theorem 4.2 to approximate the weights,

$$\begin{aligned}\widehat{\mathbb{P}(Y_B=1)} &= 1-\prod_{j\leq k}\left(1-f(x_j)\right)^{w_j} \approx 1-\prod_{j\leq k}\left(1-f(x_j)\right)^{\frac{S\left(\frac{\rho_f(x_j))}{k-1}\right)}{2k\cdot\Phi_0\left(\frac{1}{2k}\right)}}\\ &= 1-\exp\left(\sum_{j\leq k}\frac{S\left(\frac{\rho_f(x_j)}{k-1}\right)}{2k\cdot\Phi_0\left(\frac{1}{2k}\right)}\cdot\log(1-f(x_j))\right)\\ &\approx \underbrace{1-\exp\left(\int_0^1\frac{S\left(\frac{\rho_f(x))}{k-1}\right)}{2k\cdot\Phi_0\left(\frac{1}{2k}\right)}\log(1-f(x))df(x)\right)}_{T}\end{aligned}$$

where we use the exponential of the logarithm formulation in the second line, and we approximated the sum with the integral (third line), for large values of k (hypothesis). The notation $\rho_f(x)$ indicates the ranking of an instance probability $f(x)$ that is moving along the interval $[0,1]$. Then, we prove two inequalities:

$$\begin{aligned}T &< 1-\exp\left(-\frac{S\left(1\right)}{2k\cdot\Phi_0\left(\frac{1}{2k}\right)}\right) < 1 \quad \text{for large } k,\\ T &> 1-\exp\left(\frac{S\left(0.5\right)\cdot\log(1)}{2k\cdot\Phi_0\left(\frac{1}{2k}\right)}\right) > 0 \quad \text{for large } k,\end{aligned}$$

where in the first line we take the maximum rank value $\max_{t\in[0,1]} S\left(\frac{\rho_f(t)}{k-1}\right)$ and solve the integral (equal to -1), while, in the second line, we take the minimum values $\min_{t\in[0,1]} S\left(\frac{\rho_f(t)}{k-1}\right)$ and $\min_{t\in[0,1]}\log(1-t)$ and the integral gets constant equal to 1. This proves that $0<\widehat{\mathbb{P}(Y_B=1)}<1$ for any bag B for large values of k. Moreover, taking the limit on both sides it is straightforward that $\widehat{\mathbb{P}(Y_B=1)} \to l \in (0,1)$ when $k \longrightarrow +\infty$. □

For example, if the probabilities $f(x)$ were uniformly distributed over the instances in the bag, the bag probability would converge to 0.78. As a result, because the weighted noisy-OR returns non-constant bag probability estimates, varying the model parameters changes the bag probabilities, and, in turn, the gradient of the loss L_p cannot be null. This confirms that the model can use back-propagation to learn, with no constraint on the bag size k.

5 EXPERIMENTS

We address the following five experimental questions:

Q1. How does PUMA's bag and instance level performance compare to existing approaches?
Q2. How does PUMA's performance vary upon changing the number of true anomalies in a bag?
Q3. How does changing the number of reliable negatives $|R|$ impact PUMA's performance?
Q4. How does increasing the number of instances per bag k impact PUMA's performance?
Q5. How robust is PUMA to the presence of anomalies in the unlabeled data?

5.1 Experimental Setup

Methods. We compare PUMA[1] to eight baselines. The inexact AE (iAE) is the state-of-the-art competitor for multi-instance anomaly detection [19]. Because iAE needs negative bag labels, we use our own approach to select R reliable negatives for a fair comparison. Note that iAE is able to leverage bag-level labels to learn, as it is a natural MIL approach. puIF leverages the absence of negative labels by calibrating the bag probabilities of an unsupervised IForest [22] through a proper logistic regression technique for PU data [20]. For completeness, we include cIF which uses the traditional logistic calibration [31] to calibrate IForest's bag probabilities using positive and reliable negative bag labels. Moreover, a random forest classifier RF (naively) considers the unlabeled bags as negatives, and sets the instance labels to be the same as their bag label, as traditionally done in the literature [19]. Because none of the last three methods naturally link bags to instance labels, we use our own weighted noisy-OR for a fair comparison. Finally, we include four existing baselines that only assign bag probabilities: PU-SKC [2] is based on empirical risk minimization, PUmil [42] is an SVM-based approach, LSDD [38] and DSDD [11] are density-based methods.

Data. Our experiments focus on how anomaly detection can impact real-world sustainability. Specifically, we look at preventing blade icing in wind turbines and preventing water loss in retail stores. We complement these tasks, which naturally fit with this paper's setting, with additional experiments on benchmark datasets.

The first task aims at detecting anomalous ice formation on the blades of two wind turbines (T15 and T21) [45].[2] The datasets contain sensor measurements collected over the course of three months. An instance is a 10-minute contiguous segment of the data, while a bag includes 12 consecutive instances grouped together in a 2-hour long segment. This is a natural interpretation as predictions are sensibly interpreted every 10 minutes, while the ground-truth labels are reliably available at least every 2 hours.

[1] Code available at https://github.com/Lorenzo-Perini/PU-MIL-AD.

[2] The datasets can be downloaded from http://www.industrial-bigdata.com/Data

Table 1: The number of instances, bags, instances per bag k, variables, the proportion of positive instances I_Y, and of positive bags B_Y for each considered dataset.

Dataset	# Instances	# Bags	k	# Vars	I_Y	B_Y
Store1	12000	1000	12	11	0.037	0.245
Store2	12000	1000	12	11	0.048	0.341
Store3	12000	1000	12	11	0.031	0.218
Store4	12000	1000	12	11	0.076	0.359
Store5	12000	1000	12	11	0.124	0.294
Store6	12000	1000	12	11	0.018	0.153
Store7	12000	1000	12	11	0.113	0.500
Turbine15	4392	366	12	10	0.071	0.136
Turbine21	1956	163	12	10	0.052	0.098
Annthyroid	7120	712	10	21	0.075	0.300
KDDCup99	10000	1000	10	40	0.004	0.019
PageBlock	5340	534	10	10	0.081	0.300
SpamBase	2720	272	10	57	0.072	0.300
Waveform	3440	344	10	21	0.029	0.113
Cardio	1770	177	10	21	0.071	0.300
Cardiotoc.	1800	180	10	21	0.091	0.300
Internet	1740	174	10	1555	0.086	0.300
Landsat	5480	548	10	36	0.070	0.300
Letter	1570	157	10	32	0.050	0.229
Mammog.	7840	784	10	6	0.032	0.137
Musk	3060	306	10	166	0.031	0.133
Optdigits	5190	519	10	64	0.025	0.096
Pendigits	6870	687	10	16	0.022	0.097
Satellite	4750	475	10	36	0.075	0.300
Shuttle	10000	1000	10	9	0.070	0.300
Skin	7750	775	10	3	0.073	0.300
Pima	540	54	10	8	0.088	0.300
Thyroid	3650	365	10	6	0.025	0.106
Vowels	1450	145	10	12	0.031	0.131
Wilt	4810	481	10	5	0.053	0.212

The second task aims at detecting water leaks in 7 stores (S1, . . . , S7) of a large retail company.[3] The datasets contain water consumption measurements over the course of 3 years. An instance corresponds to an hour-long contiguous segment of measurements. This partition helps capture time-of-day effects and helps ensure that features are aggregating over normal and anomalous behavior. A bag corresponds to a full working day (8am-8pm) as the maintenance operators are typically flagging issues on a store level by inspecting the data once per day.

Following Iwata et al. [19], we test the methods on 21 *benchmark datasets commonly used for anomaly detection.*[4] Since Iwata et al. [19] unrealistically assume that positive bags contain a single anomaly, we consider a more general setting where positive bags can contain multiple anomalies. We construct such bags using a hierarchical approach: (i) we sample a bag label from a Bernoulli(0.3) random variable; (ii) if it is positive, we fill the bag with random $n \in \{0, \dots, \lceil \frac{k}{2} \rceil\}$ true anomalies and with $n - k$ normal instances (k in total); (iii) if it is negative, we simply fill it with k uniformly drawn normal instances. This yields for each dataset a set of bags as well as their ground-truth labels.

Table 1 shows the properties of the 30 anomaly detection datasets. Although for the benchmark datasets the bag labels are sampled using Bernoulli(0.3), some datasets have a proportion of positive bag labels lower than 0.3 (e.g., KDDCup99, Waveform). This is due to the limited number of anomalies available: once no more anomalies can be added (without repetition) to the bags, we fill in the remaining bags with only normal data. This creates fewer positive bag labels but allows us to use even datasets with a low proportion of anomalies.

Setup. For each combination of dataset and method, we run the following experiment: (1) we use a stratified 5 fold cross-validation to divide all the bags in the dataset into a train and test set, (2) we simulate gradually increasing the label frequency c, i.e., the percentage of labeled positive bags, starting from $c = 5\%$ and up to $c = 50\%$ (every 5%), (3) we train the method using the labeled (and unlabeled) training bags, (4) we use the trained method to predict the labels of both the test set instances and the test set bags, and (5) we compute the *area under the ROC curve* (AUC) on the instance- and bag-level. We compute the AUC because this is the standard evaluation measure in anomaly detection research [6]. To obtain robust results, we repeat each experiment 5 times and report the averages and the standard deviations. The combination of 30 datasets with 5 fold cross-validation, 5 repetitions, and 10 label frequencies yields a total of $30 \times 5 \times 5 \times 10 = 7500$ experiments. We run the experiments using one NVIDIA GeForce GTX 1080 Ti GPU.

Hyperparameters. PUMA and iAE use the same network architecture: two layers with 4 and 2 neurons each, learning rate= 0.005, 300 epochs, batch size= 64, and ReLU as activation function. We set the regularization term λ to 0.01. RF, cIF and puIF[5] use their default hyperparameters [37], and the Sklearn implementation of Differential Evolution [32] as optimizer.

5.2 Experimental Results

Q1. How does PUMA compare to the baselines? We compare PUMA to the baselines both on a bag and on an instance level.

Bag level. Figure 1 shows a fine-grained view of the results by plotting the bag-level AUC as a function of the label frequency c. Overall, PUMA achieves higher/similar (i.e., difference ≤ 0.01) AUC in around 72%, 77% and 83% of the experiments against, respectively, PU-SKC, RF and iAE. Moreover, for each experiment, we rank the methods from the best to the worst (lower is better) and report the average ranks in Table 2. The results show that PUMA always obtains the lowest (best) average rank when aggregating for each label frequency over the datasets. In addition, PUMA ranks in the top 2 positions on more than 75% of the experiments, while PU-SKC, RF and iAE do so only on around 35% of the experiments.

Instance level. Table 3 reports the instance-level AUCs and ranks averaged over all datasets (± std) for each method and for ten values of c. Overall, PUMA has the highest average AUC and lowest average rank compared to the baselines regardless of the value of the

[3] These data are proprietary and shared with the researchers under an NDA.

[4] Link: https://github.com/Minqi824/ADBench/tree/main/datasets/Classical

[5] IForest's code is available at https://pyod.readthedocs.io/en/latest/index.html

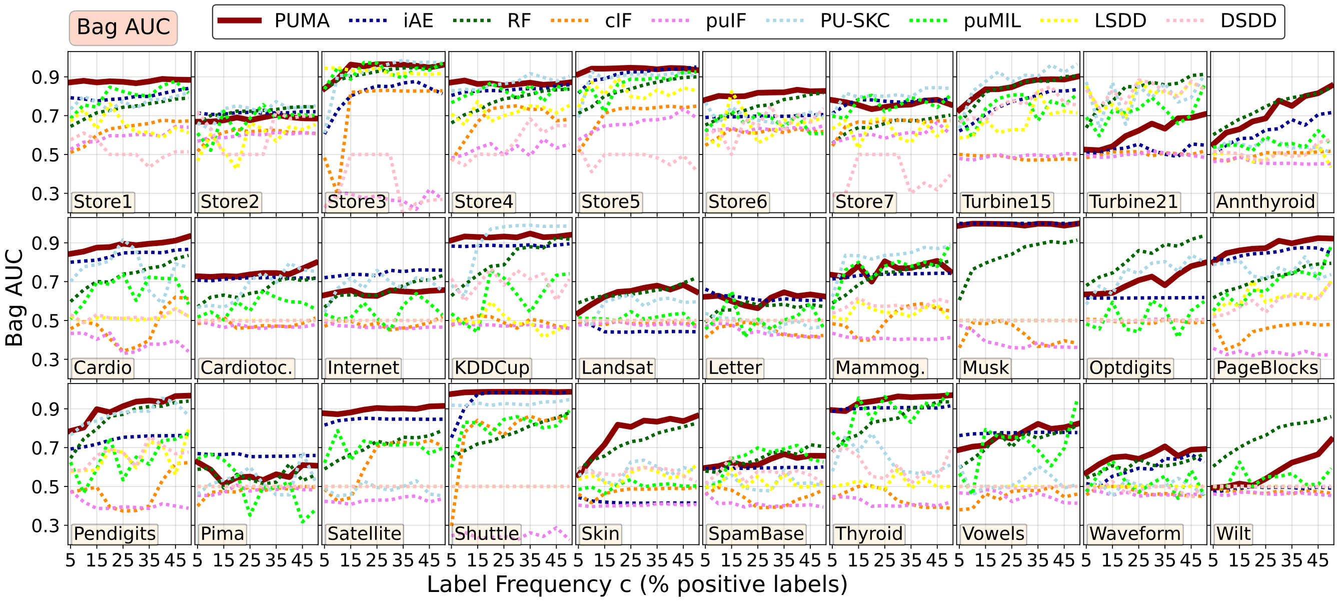

Figure 1: Bag-wise AUC obtained by PUMA and the baselines on all the 30 datasets when varying the label frequency c (x-axis). Overall, PUMA performs better/similar to the baselines on the majority of datasets.

	Bag ranks (avg. ± std.)								
c%	PUMA	iAE	RF	cIF	puIF	PU-SKC	puMIL	LSDD	DSDD
5	**1.87** ± **1.43**	3.23 ± 2.33	3.77 ± 1.59	8.13 ± 1.48	8.03 ± 1.00	3.98 ± 1.58	4.85 ± 1.81	5.68 ± 1.58	5.45 ± 1.94
10	**2.07** ± **1.31**	3.33 ± 2.17	3.40 ± 1.65	7.53 ± 1.11	8.40 ± 0.81	3.50 ± 1.86	4.97 ± 2.06	5.75 ± 1.60	6.05 ± 1.81
15	**2.17** ± **1.37**	3.63 ± 2.19	3.70 ± 1.56	7.40 ± 1.33	8.43 ± 0.73	3.48 ± 1.99	4.08 ± 2.03	5.88 ± 1.60	6.22 ± 1.97
20	**2.27** ± **1.46**	3.50 ± 2.19	3.13 ± 1.68	7.47 ± 1.36	8.30 ± 0.75	3.55 ± 1.63	4.45 ± 2.12	5.92 ± 1.57	6.42 ± 1.80
25	**2.20** ± **1.27**	3.57 ± 2.14	3.30 ± 1.51	7.43 ± 1.07	8.57 ± 0.63	3.32 ± 1.97	4.25 ± 2.04	5.92 ± 1.41	6.45 ± 1.57
30	**1.90** ± **1.27**	3.57 ± 2.08	3.13 ± 1.33	7.47 ± 1.38	8.40 ± 0.77	3.60 ± 2.15	4.43 ± 1.70	6.27 ± 1.22	6.23 ± 1.71
35	**1.97** ± **1.33**	3.60 ± 2.28	3.07 ± 1.39	7.30 ± 1.42	8.33 ± 0.76	3.72 ± 2.26	4.45 ± 1.56	6.20 ± 1.41	6.37 ± 1.68
40	**2.03** ± **1.16**	3.63 ± 2.25	2.90 ± 1.30	7.27 ± 1.17	8.43 ± 0.90	3.58 ± 2.19	4.55 ± 1.76	6.40 ± 1.28	6.20 ± 1.71
45	**1.97** ± **1.38**	3.63 ± 2.01	2.73 ± 1.36	7.33 ± 1.32	8.37 ± 0.85	3.58 ± 2.11	4.62 ± 1.55	6.35 ± 1.27	6.42 ± 1.61
50	**2.13** ± **1.38**	3.77 ± 2.14	2.77 ± 1.25	7.27 ± 1.34	8.33 ± 0.84	3.72 ± 2.07	4.55 ± 2.23	6.22 ± 1.39	6.25 ± 1.66

Table 2: Average rank (± std.) on the bag level for each method across all experiments and for ten different label frequencies. Overall, PUMA outperforms the competing baseline and always achieving always the lowest average rank.

label frequency c. Moreover, PUMA outperforms the runner-up puIF on around 73% of the experiments. As a side note, PUMA's learning curves are relatively flat, which could indicate that it mainly focuses on improving the bag-level predictions given the additional labels.

As a final remark, our approach is the only one that offers consistently good performance on both the instance and bag levels. While iAE's is competitive on the bag level, its performance drops off on the instance level. In contrast, puIF obtains competitive performance on an instance level but performs poorly on a bag level. Finally, RF achieves weak performance for low label frequencies in both scenarios. These effects might be explained by considering the purpose and assumptions made by each method. iAE assumes that each positive bag only contains one positive instance. puIF is designed to classify instances, not bags. RF assumes that unlabeled bags only contain negative instances and that all instances inside a positive bag belong to the positive class. Thus, for low frequencies, many examples are incorrectly labeled as negative. PUMA, on the other hand, is explicitly designed to handle both cases and selects the number of reliable negatives based on the number of positive bag labels.

Q2. Changing the number of ground-truth anomalies in a bag. PUMA can handle a varying number of anomalies in any given bag. However, the iAE was only evaluated in a context where each positive bag contained at most one positive instance [19]. Figure 2 (top) compares the performance of PUMA and iAE in this

	Instance AUC (avg. ± std.)					**Instance ranks** (avg. ± std.)				
c%	PUMA	iAE	RF	cIF	puIF	PUMA	iAE	RF	cIF	puIF
5	**0.75 ± 0.16**	0.72 ± 0.17	0.58 ± 0.06	0.66 ± 0.13	0.68 ± 0.14	**2.01 ± 1.06**	2.67 ± 1.49	4.10 ± 1.24	3.27 ± 1.36	2.87 ± 1.14
10	**0.77 ± 0.15**	0.74 ± 0.18	0.63 ± 0.06	0.66 ± 0.14	0.71 ± 0.14	**1.87 ± 1.04**	2.70 ± 1.53	3.80 ± 1.30	3.67 ± 1.32	2.97 ± 0.96
15	**0.78 ± 0.16**	0.74 ± 0.18	0.67 ± 0.06	0.65 ± 0.15	0.73 ± 0.15	**1.90 ± 1.12**	2.77 ± 1.45	3.63 ± 1.35	3.87 ± 1.25	2.83 ± 1.02
20	**0.79 ± 0.15**	0.75 ± 0.18	0.70 ± 0.07	0.67 ± 0.14	0.75 ± 0.16	**2.03 ± 1.17**	2.83 ± 1.56	3.53 ± 1.46	3.77 ± 1.01	2.73 ± 1.28
25	**0.80 ± 0.15**	0.75 ± 0.18	0.72 ± 0.07	0.68 ± 0.14	0.76 ± 0.15	**2.03 ± 1.03**	3.00 ± 1.58	3.47 ± 1.55	3.90 ± 1.12	2.60 ± 0.97
30	**0.81 ± 0.13**	0.75 ± 0.18	0.74 ± 0.07	0.70 ± 0.14	0.76 ± 0.16	**1.87 ± 0.90**	2.97 ± 1.54	3.50 ± 1.61	3.90 ± 0.88	2.77 ± 1.14
35	**0.81 ± 0.14**	0.76 ± 0.18	0.76 ± 0.07	0.71 ± 0.15	0.76 ± 0.15	**1.97 ± 1.07**	2.77 ± 1.57	3.33 ± 1.58	3.83 ± 1.12	3.10 ± 0.99
40	**0.82 ± 0.13**	0.76 ± 0.18	0.78 ± 0.07	0.69 ± 0.14	0.76 ± 0.15	**2.00 ± 1.11**	2.97 ± 1.45	3.10 ± 1.60	4.00 ± 1.05	2.93 ± 1.11
45	**0.82 ± 0.13**	0.76 ± 0.17	0.80 ± 0.08	0.69 ± 0.14	0.76 ± 0.15	**1.70 ± 0.88**	2.97 ± 1.47	2.93 ± 1.46	4.23 ± 0.97	3.17 ± 0.99
50	**0.83 ± 0.13**	0.77 ± 0.18	0.81 ± 0.08	0.69 ± 0.15	0.76 ± 0.15	**1.80 ± 0.81**	2.97 ± 1.47	2.87 ± 1.46	4.07 ± 1.23	3.30 ± 1.06

Table 3: Average AUC and average rank on the instance level for each method across all experiments and for ten different label frequencies. PUMA outperforms every baseline obtaining both the highest average AUC and the lowest average rank.

restricted setting on the 9 benchmark datasets used in [19]. Figure 2 (bottom) considers the more realistic scenario where a positive bag can contain multiple anomalous instances (we randomly vary the amount in $[1, \lceil \frac{k}{2} \rceil]$). When there is one anomaly per positive bag, PUMA has an overall performance similar to iAE both in terms of average AUC and ranks: it performs clearly better on Annthyroid, KDDCup, and Wilt, similar on Cardiotoc., Pageblocks, and SpamBase, and worse on Waveform, Internet, and Pima (3 datasets each case). However, in the more realistic and general setting PUMA outperforms iAE on 7 out of 9 datasets. This clearly shows how our loss function leverages multiple high probabilities on an instance level to derive the positive bag probability, as opposed to iAE, which only cares for the highest instance probability.

Q3. Impact of R on the performance of PUMA?. PUMA selects $|R|$ reliable negatives as representatives of the negative bag distribution. Our intuition is that when there are few positive bag labels (i.e., low label frequency c), identifying and including many negatives in the labeled component L_p of PUMA's loss would make the model unduly biased towards the negative class. In contrast, if the label frequency c is high and too few reliable negatives were selected, the model would be biased toward the positive class and underestimate the probability of unlabeled bags being negative. This idea is confirmed by Figure 3, which shows how PUMA's performance changes when varying the proportion of reliable negatives as a function of the label frequency c on the 7 store datasets. Hence, the heuristic of setting $|R| = |P|$, as confirmed by Basile et al. [3], is reasonable.

Q4. Impact of k on the performance of PUMA?. To assess the impact of the instance sample size k on the model performance, we run experiments on the 7 retail store datasets, as the value of k is interpretable. We vary k to get a bag as half-day ($k = 6$), a day ($k = 12$), two days ($k = 24$), three days ($k = 36$) and four days ($k = 48$). Due to the computational cost, we limit the experiments to 40 positive and 60 negative bags selected at random, and consider two representative label frequencies, $c = 10\%, 30\%$ (i.e., 4 and 12 positive labels). Figure 4 shows the results (mean ± std) for both bag and instances. Overall, the pattern is slightly decreasing for high values of k, as the model has to propagate the bag label to more instances. However, k only has a small impact on the final performance, as PUMA learns the bag probability through several instances in the bag (weighted noisy-OR) instead of simply using the max value (as does iAE).

Q5. How robust is PUMA to the presence of anomalies in the unlabeled data? Because the unlabeled loss L_u does not distinguish between normals and anomalies, we verify whether having contaminated unlabeled instances weakens the whole model. Following the traditional anomaly detection pipeline, a straightforward approach to remove the potential anomalies from the unlabeled bags is to drop off the top $t\%$ instance scores before computing L_u. By removing those instances during training, the autoencoder does not improve its reconstruction of such potential anomalies, but only learns how to reconstruct the normal patterns.

We experimentally test our hypothesis that dropping potential anomalies could improve PUMA's performance. We fix $t\%$ to be $[1\%, 5\%, 10\%]$ and compare these three new baselines with our original version, running the same experiments as before. On a bag level, the three baselines obtain, respectively, an average rank of $2.14, 2.77, 3.06$ while our original version gets 2.02 (all std. around 1). Similarly, on an instance level, the baselines achieve an average rank of $2.34, 2.58, 2.99$ while the original version gets 2.09 (all std. around 1). Because there is no gained benefit to forcing the model to ignore part of the unlabeled instances, we conclude the method is quite robust to unlabeled anomalous instances.

6 CONCLUSION

In this paper, we tackled the problem of learning from positive and unlabeled bags in anomaly detection, where a bag is a set of instances. We proposed PUMA, a method that assigns instance and bag probabilities by learning through a two-components loss function. Via the unlabeled loss component L_u, PUMA uses the unlabeled bags to learn a representation of the seen examples in order to detect novelties at the test time. Via the labeled loss component L_p, it uses the positive bags to learn (possible) anomalous patterns in four steps. First, it parametrizes the instance probability function by transforming the autoencoder's anomaly scores into probabilities through a sigmoid function (Platt scaling). Second,

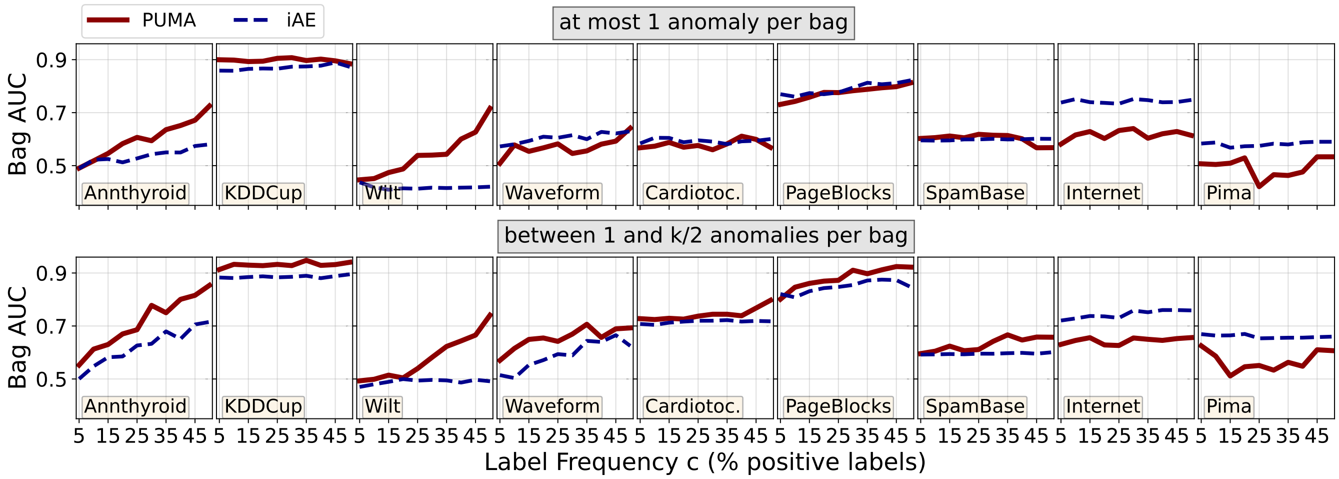

Figure 2: Comparison between PUMA’s and ɪAE’s performance on the restrictive Iwata et al. [19] setting where each positive bag must contain exactly one anomaly (top) and the more realistic setting where bags may contain multiple anomalies (bottom). While in the restricted scenario (top) the two methods perform similarly, in the more realistic general scenario PUMA clearly outperforms ɪAE on 7 out of 9 datasets (bottom).

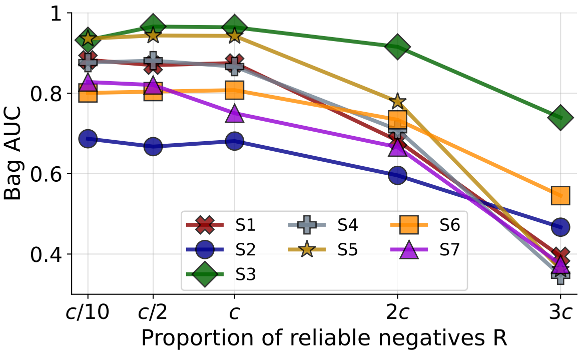

Figure 3: PUMA’s bag AUC when varying the percentage of reliable negatives as a function of c. PUMA’s performance drops when the percentage of reliable negatives is higher/lower than c.

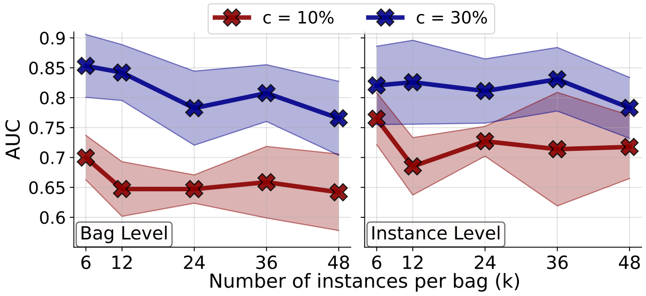

Figure 4: PUMA’s AUC on a bag (left) and an instance (right) level for two label frequencies (10%, 30%) when varying the bag size k. The shade represents the uncertainty (standard deviation) due to multiple experiments on multiple datasets. Overall, PUMA’s performance is only weakly affected by the bag size.

it builds a bridge between the instance and the bag levels by using the weighted Noisy-OR, which derives the bag probabilities by aggregating the instance probabilities with different weights. The key insight to setting proper weights is that the instances with the highest and the lowest probabilities in a bag should contribute more to defining the bag label. Third, PUMA self-generates negative bag labels by labeling the bags with the lowest positive probability as negative. Finally, it measures the log-likelihood of the given labels under the estimated bag probabilities and uses its negative value as L_p. We theoretically showed that PUMA learns from this setting even for large bag size k. Empirically, we evaluated our method on 21 benchmarks and 9 real-world datasets and compared it to several baseline methods. Experimental results show that PUMA performs better than all the baselines both on an instance and a bag level on the majority of the datasets.

ACKNOWLEDGEMENTS

This work is supported by the FWO-Vlaanderen (aspirant grant 1166222N to LP and G0D8819N to JD), the Flemish government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme (JD, LP), the “Agentschap Innoveren & Ondernemen (VLAIO)" as part of the innovation mandate HBC.2020.2297 (VV), and the KU Leuven Research Fund (3E211302) (VV).

REFERENCES

[1] Pierre Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR

Workshop and Conference Proceedings, PMLR, Bellevue, Washington, USA, 37–49.

[2] Han Bao, Tomoya Sakai, Issei Sato, and Masashi Sugiyama. 2018. Convex formulation of multiple instance learning from positive and unlabeled bags. *Neural Networks* 105 (2018), 132–141.

[3] Teresa Basile, Nicola Di Mauro, Floriana Esposito, Stefano Ferilli, and Antonio Vergari. 2017. Density estimators for positive-unlabeled learning. In *International Workshop on New Frontiers in Mining Complex Patterns*. Springer, Springer International Publishing, Cham, 49–64.

[4] Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning* 109 (2020), 719–760.

[5] Forrest Briggs, Xiaoli Z Fern, and Raviv Raich. 2012. Rank-loss support instance machines for MIML instance annotation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, NY, USA, 534–542. https://doi.org/10.1145/2339530.2339616

[6] Guilherme O Campos, Arthur Zimek, Jörg Sander, Ricardo JGB Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E Houle. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery* 30 (2016), 891–927.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[8] Sneha Chaudhari and Shirish Shevade. 2012. Learning from positive and unlabelled examples using maximum margin clustering. In *Neural Information Processing: 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part III 19*. Springer, Springer Berlin Heidelberg, Berlin, Heidelberg, 465–473.

[9] Francesco Del Buono, Francesca Calabrese, Andrea Baraldi, Matteo Paganelli, and Francesco Guerra. 2022. Novelty detection with autoencoders for system health monitoring in industrial environments. *Applied Sciences* 12, 10 (2022).

[10] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence* 89, 1-2 (1997), 31–71.

[11] Marthinus Christoffel Du Plessis, Gang Niu, and Masashi Sugiyama. 2013. Clustering unclustered data: Unsupervised binary labeling of two datasets having different class balances. In *2013 Conference on Technologies and Applications of Artificial Intelligence*. IEEE, 1–6.

[12] Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, NY, USA, 213–220.

[13] Abdelali Elmoufidi, Khalid El Fahssi, Said Jai-andaloussi, Abderrahim Sekkaki, Quellec Gwenole, and Mathieu Lamard. 2018. Anomaly classification in digital mammography based on multiple-instance learning. *IET Image Processing* 12, 3 (2018), 320–328.

[14] Songhe Feng and De Xu. 2010. Transductive multi-instance multi-label learning algorithm with application to automatic image annotation. *Expert Systems with Applications* 37, 1 (2010), 661–670.

[15] Jun Han and Claudio Moraga. 1995. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *International workshop on artificial neural networks*. Springer, Springer Berlin Heidelberg, Berlin, Heidelberg, 195–201.

[16] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. 2022. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems* 35 (2022), 32142–32159.

[17] Jing Huo, Yang Gao, Wanqi Yang, and Hujun Yin. 2012. Abnormal event detection via multi-instance dictionary learning. In *International conference on intelligent data engineering and automated learning*. Springer, Springer Berlin Heidelberg, Berlin, Heidelberg, 76–83.

[18] Jing Huo, Yang Gao, Wanqi Yang, and Hujun Yin. 2014. Multi-instance dictionary learning for detecting abnormal events in surveillance videos. *International journal of neural systems* 24, 03 (2014), 1430010.

[19] Tomoharu Iwata, Machiko Toyoda, Shotaro Tora, and Naonori Ueda. 2020. Anomaly detection with inexact labels. *Machine Learning* 109 (2020), 1617–1633.

[20] Kristen Jaskie, Charles Elkan, and Andreas Spanias. 2019. A Modified Logistic Regression for Positive and Unlabeled Learning. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2007–2011. https://doi.org/10.1109/IEEECONF44664.2019.9048765

[21] Weixin Li and Nuno Vasconcelos. 2015. Multiple instance learning for soft bags via top instances. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[22] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422. https://doi.org/10.1109/ICDM.2008.17

[23] Guoqing Liu, Jianxin Wu, and Zhi-Hua Zhou. 2012. Key instance detection in multi-instance learning. In *Asian Conference on Machine Learning*. PMLR.

[24] Oded Maron and Tomás Lozano-Pérez. 1997. A framework for multiple-instance learning. *Advances in neural information processing systems* (1997).

[25] Lorenzo Perini, Paul Bürkner, and Arto Klami. 2023. Estimating the Contamination Factor's Distribution in Unsupervised Anomaly Detection. In *International Conference on Machine Learning*. PMLR.

[26] Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. 2020. Class prior estimation in active positive and unlabeled learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence*. IJCAI-PRICAI.

[27] Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. 2020. Quantifying the confidence of anomaly detectors in their example-wise predictions. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020*. Springer, 227–243.

[28] Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. 2022. Transferring the contamination factor between anomaly detection domains by shape similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4128–4136.

[29] Sang Phan, Duy-Dinh Le, and Shin'ichi Satoh. 2015. Multimedia event detection using event-driven multiple instance learning. In *Proceedings of the 23rd ACM international conference on Multimedia*.

[30] John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* (1999).

[31] John C Platt. 2000. 5 Probabilities for SV Machines. *Advances in Large Margin Classifiers* (2000).

[32] Kenneth V Price. 2013. Differential evolution. In *Handbook of optimization*. Springer.

[33] Gwenolé Quellec, Guy Cazuguel, Béatrice Cochener, and Mathieu Lamard. 2017. Multiple-instance learning for medical image and video analysis. *IEEE reviews in biomedical engineering* (2017).

[34] Gwenolé Quellec, Mathieu Lamard, Michel Cozic, Gouenou Coatrieux, and Guy Cazuguel. 2016. Multiple-instance learning for anomaly detection in digital mammography. *Ieee transactions on medical imaging* (2016).

[35] Leonard Sabetti and Ronald Heijmans. 2021. Shallow or deep? Training an autoencoder to detect anomalous flows in a retail payment system. *Latin American Journal of Central Banking* (2021).

[36] Beomjo Shin, Junsu Cho, Hwanjo Yu, and Seungjin Choi. 2021. Sparse Network Inversion for Key Instance Detection in Multiple Instance Learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*.

[37] Jonas Soenen, Elia Van Wolputte, Lorenzo Perini, Vincent Vercruyssen, Wannes Meert, Jesse Davis, and Hendrik Blockeel. 2021. The Effect of Hyperparameter Tuning on the Comparative Evaluation of Unsupervised Anomaly Detection Methods. In *Proceedings of the KDD'21 Workshop on Outlier Detection and Description*. Outlier Detection and Description Organising Committee.

[38] Masashi Sugiyama, Takafumi Kanamori, Taiji Suzuki, Marthinus Christoffel Du Plessis, Song Liu, and Ichiro Takeuchi. 2013. Density-difference estimation. *Neural Computation* 25, 10 (2013), 2734–2775.

[39] Vincent Vercruyssen, Lorenzo Perini, Wannes Meert, and Jesse Davis. 2023. Multi-domain Active Learning for Semi-supervised Anomaly Detection. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022*. Springer, 485–501.

[40] Vincent Vercruyssen, Meert Wannes, Verbruggen Gust, Maes Koen, Bäumer Ruben, and Davis Jesse. 2018. Semi-supervised anomaly detection with an application to water analytics. In *Proceedings of 18th IEEE International Conference on Data Mining*. IEEE.

[41] Junxiang Wang, Liang Zhao, and Yanfang Ye. 2018. Semi-supervised multi-instance interpretable models for flu shot adverse event detection. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE.

[42] Jia Wu, Xingquan Zhu, Chengqi Zhang, and Zhihua Cai. 2014. Multi-instance learning from positive and unlabeled bags. In *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part I 18*. Springer, 237–248.

[43] Xin-Shun Xu, Yuan Jiang, Xiangyang Xue, and Zhi-Hua Zhou. 2012. Semi-supervised multi-instance multi-label learning for video annotation task. In *Proceedings of the 20th ACM international conference on Multimedia*.

[44] Cha Zhang, John Platt, and Paul Viola. 2005. Multiple instance boosting for object detection. *Advances in neural information processing systems* (2005).

[45] Lijun Zhang, Kai Liu, Yufeng Wang, and Zachary Bosire Omariba. 2018. Ice detection model of wind turbine blades based on random forest classifier. *Energies* (2018).

[46] Hongshan Zhao, Huihai Liu, Wenjing Hu, and Xihui Yan. 2018. Anomaly detection and fault analysis of wind turbine components based on deep learning network. *Renewable energy* (2018).

[47] Yu Zhou and Anlong Ming. 2016. Semi-Supervised Multiple Instance Learning and its application in visual tracking. In *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*. IEEE.

[48] Zhi-Hua Zhou. 2004. Multi-instance learning: A survey. *Department of Computer Science & Technology, Nanjing University, Tech. Rep* (2004).

[49] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. 2009. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*.