This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2871421, IEEE Transactions on Multimedia

Boosting Positive and Unlabeled Learning for Anomaly Detection with Multi-features

Jiaqi Zhang, Zhenzhen Wang, Jingjing Meng, Yap-Peng Tan, *Senior Member, IEEE* and Junsong Yuan, *Senior Member, IEEE*

***Abstract*—One of the key challenges of machine learning-based anomaly detection relies on the difficulty of obtaining anomaly data for training, which is usually rare, diversely distributed, and difficult to collect. To address this challenge, we formulate anomaly detection as a Positive and Unlabeled (PU) learning problem where only labeled positive (normal) data and unlabeled (normal and anomaly) data are required for learning an anomaly detector. As a semi-supervised learning method, it does not require to provide labeled anomaly data for the training, thus is easily deployed to various applications. As the unlabeled data can be extremely unbalanced, we introduce a novel PU learning method which can tackle the situation where unlabeled data set is mostly composed of positive instances. We start by using a linear model to extract the most reliable negative instances followed by self-learning process to add reliable negative and positive instances with different speeds based on the estimated positive class prior. Furthermore, when feedback is available, we adopt boosting in the self-learning process to advantageously exploit the instability characteristic of PU learning. The classifiers in self-learning process are weighted combined based on the estimated error rate to build the final classifier. Extensive experiments on six real datasets and one synthetic dataset show that our methods have better results under different conditions compared to existing methods.**

***Index Terms*—Anomaly Detection, PU Learning, Semi-supervised Learning, Boosting.**

I. Introduction

ANOMALY detection is a very important topic in security, economics and medical fields, it involves different types of multimedia data structures. For example, surveillance and monitoring systems that employ a large number of cameras to capture people's activities in the environment (video) [1]; recorded acoustic surveillance of an open public space (audio) [2]; heterogeneous categorical events of enterprise computer system (text and number) [3]. The main challenges of anomaly detection comparing to traditional classification problems are

- Availability of labeled data for training is a major issue. It is difficult to get labeled anomaly instances with various behaviors. Thus, most of the data are unlabeled.
- The proportion of anomalies is usually much smaller than normal data, which makes it an unbalanced problem.
- The inter-class similarity and intra-class variation make it difficult to separate the normal and anomalous behaviors.

J. Zhang was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 642273 e-mail: jzhang069@e.ntu.edu.sg.

Z. Wang, J. Meng, Y. Tan and J. Yuan are with Nanyang Technological University.

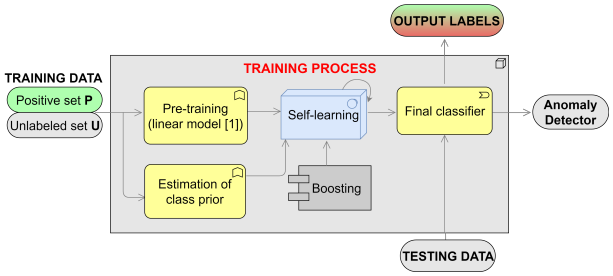


Fig. 1: Structure for the proposed method.

- It is not known in advance which features are more discriminative in identifying anomalies with the multi-feature data structure.

The challenges can be tackled by Positive and Unlabeled (PU) learning, which is a kind of semi-supervised learning that only labeled positive samples and unlabeled samples are required in the training process. The characteristic of PU learning is that no labeled negative instances are required and the anomalies are regarded as the negative class to avoid labeling of anomalies. When both positive data and unlabeled data are available, instead of using part of the provided data as unsupervised and supervised one-class classification methods, PU learning can make full use of the provided data. In addition, PU learning can avoid labeling anomalies comparing to supervised two-class classification. Many previous studies also propose methods for feature selection [4] while some machine learning methods are naturally capable of handling multi-features data distribution like a linear model and Random Forest.

PU learning has many applications such as text detection [5], [6], [7], [8] and disease gene identification [9], [10], [11]. However, the common practice for existing PU learning methods is to assume that the unlabeled set is dominated by negative instances. Based on this assumption, conventional PU learning methods usually start training the initial classifier by regarding all the unlabeled data as negative [5], [6]. Then, positive instances in the unlabeled set are contamination for those methods. Thus higher ratio of positive instances in the unlabeled set will lead to a higher error rate. In the literature, the proportion of positive instances in the unlabeled set is termed as *positive class prior*. Given positive samples and a set of unlabeled samples, Du Plessis *et al.* [12], [13] introduce an unconstrained linear model which is trained under multiple cost functions to discover anomalies. However, if more positive data and negative data act similarly, the linear

model cannot have a satisfactory separation at the confusing boundary of two classes. To improve the robustness of our method, we introduce a self-learning process, which works by iteratively extracting reliable positive and negative instances from the unlabeled set.

Another problem for PU learning is that the structure of PU learning is not very stable. Due to the lack of labels of unlabeled set U, the self-learning process cannot be sure to improve the previous step performance. Thus, more iterations of self-learning may not improve the performance of the final classifier. To address the instability of the self-learning process, we exploit a boosting-like procedure during the self-learning since boosting can improve the performance of individual classifiers. When feedback is available, we estimate the error rate to get the weight for the classifier in each self-learning step. The boosting structure in the self-learning process is like a memory device, and the weight of each classifier is to decide how much we should memorize from each step.

In this paper, we first introduce a new PU learning framework, so that conventional PU learning method can be applied to anomaly detection problem (Section III (A)) [14]. Then we propose a more powerful framework by integrating PU learning process with boosting procedure to tackle the instability of self-learning (Section III (B)). The main structure of the proposed method is illustrated in Fig. 1. Firstly we use a linear model [13] with different cost functions for positive and unlabeled set to obtain the most reliable negative instances. Then in the self-learning process, we train a classifier from scratch based on the updated reliable negative set and positive set in each step. Reliable positive and negative instances are gradually extracted with relative speeds based on the prediction results of the current step classifier. The positive class prior is estimated to adjust different learning speeds for positive and negative class. To narrow the error of positive class prior estimation, the remaining unlabeled instances will be directly classified by the classifier in the last step, rather than dividing based on estimated positive class prior. The pipeline for the proposed PU learning method raised above is shown in Section III (A) and summarized in Fig. 2. When feedback is available in the self-learning process, we integrate the PU learning process with boosting procedure. The classifiers are combined by a weighted sum of weak classifiers from each step of self-learning. The weight of each classifier depends on the estimated error rate of each step. Reliable instances are collected based on the prediction results of combined classifiers. A theoretical proof of the selection of weights is given in Section III (B). The pipeline for the updated boosting PU learning framework is illustrated in Fig. 3.

Extensive experiments on six real-world datasets and one synthetic dataset are conducted under different ratios of positive set in training data, different positive class priors and different overlapping ratios of two classes. The experimental results show that the proposed methods outperform existing methods in different settings. To summarize, our main contributions are three folds:

- We propose to use Positive and Unlabeled (PU) learning in anomaly detection problem to avoid labeling of anomalies and make full use of the training data.
- We introduce a new PU learning framework for an uncommon unbalanced scenario where the unlabeled set is dominated by positive data.
- To tackle the instability of self-learning process, we are motived by the characteristic of boosting to optimize the classifiers by dynamic weights when feedback is obtained.
- To our best knowledge, we are the first to combine PU learning and boosting, and apply to anomaly detection problem.

The rest of the paper is organized as follows: Section II briefly describes related works. The proposed frameworks are explained in Section III (A) and III (B). Experiment settings and results are presented in Section IV. Section V concludes the paper.

II. Related Works

Anomaly Detection. There are plenty of works on anomaly detection problem using supervised and unsupervised methods. These techniques can be roughly divided into following categories: classification based [15], [16], [17], clustering based [18], nearest neighbor based [19], statistical [3], information theoretic [20] and spectral [21]. Zhou *et al.* [15] propose to use Convolutional Neural Networks (CNN) for video anomaly detection. To ensure the robustness to local noise, they select the spatial-temporal volumes with a large portion of moving pixels to the spatial-temporal convolutions. Xiao *et al.* [16] propose to use Bayesian Model Averaging (BMA) by averaging over the k-best Naive Bayes (BN) classifiers. Ghosh *et al.* [19] introduce two different types of anomalies which may occur during crawling of search engines and two novel similarity measures based on vertex neighborhood and signature to overcome both conventional and the proposed anomalies. Chen *et al.* [3] directly model the likelihood of heterogeneous categorical events. It is achieved by weighted pairwise interactions among entities that are quantified based on entity embeddings. Noise-contrastive estimation is utilized to reduce the computational complexity so that the model can be learned efficiently regardless of the exponentially large event space. Ntalampiras *et al.* [2] explore novelty detection of acoustic surveillance by a probabilistic model that captures diverse characteristics and feeds the feature coefficients to three probabilistic novelty detection methodologies. Hu *et al.* [22] propose a cost-sensitive and a multi-perspective method by combining multi-instance learning (MIL) with sparse coding to recognize sensitive video with visual emotional features from videos. Alippi *et al.* [23] detect anomalies in data streams from multi-sensor units. A Hidden Markov Model (HMM) extracting the pattern structure is applied on features modeled by linear dynamic time-invariant models to detect the changes along the time series.

Semi-Supervised Learning. Anomaly detection can be treated as a classification problem under extremely unbalanced data distribution. Motivated by the fact that unlabeled data can help with the correlations among multi-modalities and be easily collected, many semi-supervised methods are exploited

for classification tasks. Ahn *et al.* [24] propose to combine semi-supervised spectral clustering based on modified pairwise constraints with multiple segmentation schemes to build a face and hair region labeler. Jian *et al.* [25] propose semi-supervised bi-dictionary learning for image classification by constructing smoothed reconstruction for discriminative dictionary and producing reconstruction coefficients in the feature space. Wu *et al.* [26] introduce a cross-task network composed of two streams: clustering task to explore the image data structure by pairwise constraints from unlabeled image; and classification task associated with a weighted softmax loss. The deep model is gradually trained by a self-paced learning paradigm. The unlabeled data are weighted to alleviate the influence of ambiguous data on model training. Zhang *et al.* [27] propose generalized semi-supervised structured subspace learning method for the task of cross-modal retrieval.

PU Learning. However, to our best knowledge, no previous work has applied PU learning to anomaly detection problem to avoid manually labeling of anomalies. Many studies on PU learning have been conducted in the past for the situation that unlabeled set is dominated by the negative data. Liu *et al.* [5] first regard the unlabeled set as negative set to get the initial classifier. Then, spy positive samples are introduced in the unlabeled set to get the reliable negative instances. Finally, Expectation-Maximization (EM) is used to build the classifier by using the positive samples and the extracted negative samples. Li *et al.* [6] propose an effective technique, which combines Rocchio method and Support Vector Machine (SVM) for classifier building. The first step is also to regard all unlabeled samples as negative and use Rocchio classifier with clustering to identify the reliable negative samples. Most of the other papers propose their methods based on [5], [6]. [8] firstly identify some negative instances using similar way as [6] from unlabeled set and generate some representative positive examples and negative examples based on Latent Dirichlet Allocation. Two similarity weights are assigned to the remaining unlabeled instances and incorporated into modified SVM to build the final classifier. Li *et al.* [7] first use spy algorithm, then use EM algorithm with Naive Bayes classifier to get final labels, which is very similar to [5]. Xiao *et al.* [28] introduce a new way to tackle the remaining unlabeled samples or called ambiguous examples of spy technique. Rather than excluding them from the training phase or simply enforcing them to either class, they associate the ambiguous examples with two similarity weights, which concern both local and global information, to build an SVM based final classifier. Nigam *et al.* [29] also use the combination of EM and NB classifier. They first use available labeled data to probabilistically label the unlabeled data, then use the labels to iteratively train new classifiers until it converges. Xia *et al.* [30] first use spy technique to select the samples with higher in-target-domain probability. Calibrated in-target-domain probabilities are used as sampling weights for training an instance-weighted NB model. Fusilier *et al.* [31] have a more conservative algorithm for extracting the reliable negative instances. They first extract the negative instances using the original PU learning method, then apply the classifier to those extracted negative instances until the number of reliable negative instances is smaller than labeled positive instances.

Plessis *et al.* [13] propose to use a linear model with different cost functions to classify. This method will not be constrained by the positive class prior but they assume positive class prior is given. High error rate will be induced when two classes have higher overlapping. Mordelet *et al.* [32] iteratively train many binary classifiers that can discriminate the known positive examples from random subsamples of the unlabeled set, and average their predictions. The method combines PU learning with the idea of bagging to discriminate positive set with unlabeled set. If by chance the unlabeled set is mostly composed of negative examples, i.e., has low contamination by positive examples, then it will probably obtain a better classifier than if it contains mostly positive examples, i.e., has high contamination. Hence, it is not appropriate in the case of anomaly detection, where positive samples dominate the unlabeled set.

III. The Proposed Techniques

Let $X = \{x_1, ..., x_i, ..., x_n\} \in R^{m \times n}$ denote the training data. $Y = \{y_1, ..., y_i, ..., y_n\} \in \{-1, 1\}$ is a n-dimension vector where $y_i = 1$ if x_i is positive and $y_i = -1$ if x_i is negative. In PU learning, Y is partially provided. Labeled positive set is denoted as $P = \{x_i \mid y_i = 1\}$ and unlabeled set is denoted as U. The positive class prior π is the portion of positive instances $\pi_0 = p(Y = 1)$ in the training set, which can be represented by $\frac{\text{number of positive samples in } U}{\text{number of samples in } U}$. The distribution of unlabeled set is

$$q(X, \pi) = \pi p(X \mid Y = 1) + (1 - \pi) p(X \mid Y = -1). \quad (1)$$

Since positive class prior is not provided for anomaly detection problem, we will use estimated positive class prior π for the following equations. We first use the linear model [13] as the pre-training process. It utilizes different loss functions for positive and unlabeled set. The relationship between the unlabeled set and positive set is studied to derive different loss functions for P and U. The self-learning process is introduced to gradually extract reliable positive and negative instances from U and add into the corresponding classes. The estimated positive class prior is used as the relative learning speed of the two classes. The self-learning process is stopped before the error of the estimated positive class prior has an influence on the classification accuracy of self-learning process. The final classifier is built by dividing the remaining unlabeled instances based on the predicted labels instead of dividing according to the estimated positive class prior. The updated boosting PU learning framework will be introduced in section III (B). It has a different self-learning process, where reliable instances are obtained based on the weighted combined classifiers where each classifier is obtained from each self-learning step. The weight is obtained from the estimated error rate of each classifier if feedback is provided during the self-learning process.

A. Proposed PU learning framework

1) Pre-training using Linear Model: We are only provided with labeled positive set and unlabeled set, thus, the first step

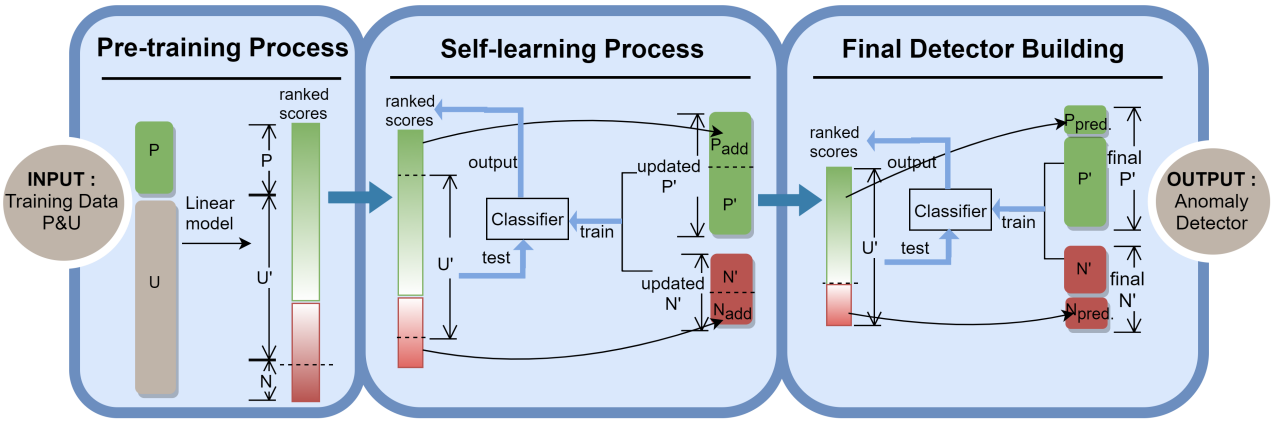


Fig. 2: Framework for proposed PU learning method. We first apply a linear model [13] to labeled positive set P and unlabeled set U to find the most reliable negative instances. In the self-learning process, the classifier of each step is trained by updated positive set P' and negative set N'. Subsequently, the classifier is applied to the remaining unlabeled set U' to predict reliable negative and positive instances and add them to the training pool. Finally, the remaining unlabeled instances are divided based on the predicted labels and the final classifier is built.

is to extract the most reliable negative set N from the unlabeled set U. For conventional PU learning methods, positive data are the detection target and the unlabeled instances are mostly negative data. Thus, the traditional methods usually regard unlabeled data as negative to build the initial classifier. However, for anomaly detection problem, labeled positive instances are normal data and anomalies are still the minority in unlabeled set. Thus, conventional PU learning methods that regard unlabeled data as negative data cannot be applied to anomaly detection problem directly.

To address the above problem, we use a linear model $g = a^T\psi(x) + b$ and apply different cost functions for positive set P and unlabeled set U [13]. Here $\psi(X) = [\psi_1(X)\dots\psi_m(X)]$ is a set of basic functions. Since there are only two classes (positive and negative) in training set and the positive class prior is estimated, the expected misclassification rate $J(g)$ can be expressed as

$$J(g) = \pi E_1[l(g(U))] + (1-\pi)E_{-1}[l(-g(U))] \tag{2}$$

when g is applied to unlabeled sample distribution. Here l is the loss function for unlabeled set. The objective function of the linear model can be simply expressed as

$$\min_{g\in G} J(g). \tag{3}$$

Due to the data distribution of training data as Eq(1), we have

$$E_U[l(-g(U))] = \pi E_1[l(-g(U))] + (1-\pi)E_{-1}[l(-g(U))]. \tag{4}$$

Substitute Eq(3) into Eq(4), the expected misclassification rate of training data can be represented by positive data and unlabeled data [13]

$$J(g) = \pi E_1[l(g(U)) - l(-g(U))] + E_U[l(-g(U))]. \tag{5}$$

Here a composite loss $\tilde{l}(z)$ can be defined as $\tilde{l}(z) = l(z) - l(-z)$ for positive set. Loss function $l(z)$ is chosen as squared loss $l_s(z) = \frac{1}{4}(z-1)^2$ so that the objective function can be analytically solved. After choosing loss function $l(z)$ as squared loss for unlabeled set, the composite loss for positive set $\tilde{l}(z) = -z$, so that misclassification rate of training data can be further simplified as [13]

$$J_s(g) = \pi E_1[-g(U)] + E_U[l_s(-g(U))]. \tag{6}$$

The scores obtained from the linear model indicate the probability of being positive. We rank the scores for instances in unlabeled set U. Those with higher scores have a high probability to be positive, while those occupy low rank on the list are more likely to be negative data. Thus, we include $h = s\% \times$ (no. of negative instances in U) instances with the highest score to the reliable negative set N where $s\%$ is a small ratio to keep extracted negative instance reliable. Since $\pi = \frac{\text{number of positive samples in } U}{\text{number of samples in } U}$, the final number of added reliable instances will be

$$h = s\% \times \mid U \mid \times(1-\pi).$$

At the same time, those instances are removed from unlabeled set U and set U is updated as U'.

Due to the real situation of anomaly detection applications, we usually do not know the proportion of anomalies in data population. The real positive class prior π_0 is assumed to be unknown. The positive class prior π is estimated based on f-divergence

$$\pi = arg\min_{0\le\pi\le1}\int f(\frac{q(x;\pi)}{p(x)})p(x)dx \tag{7}$$

with penalized function[33]

$$f(z) = \begin{cases} \mid z-1 \mid & 0 \le z \le 1, \\ \infty & \text{otherwise.} \end{cases} \tag{8}$$

Partial model

$$q(x,\pi) = \pi p(x \mid y=1) \tag{9}$$

is used to tackle the data structure of PU learning problem considering the lack of labeled negative instances.

The parameter a of linear model indicates how discriminative of each feature. Those with high value of a are more

important in the detection task. Due to the characteristic of linear model, a leads to better performance if multi-features are provided from the datasets. At the end of the first step by linear model, we have the provided labeled positive set P, the extracted reliable negative set N' and updated unlabeled set U'. Labeled instances from both classes are available, thus self-learning process can be applied in the next step.

Algorithm 1 Self-learning process

Input: Training set: P, U', N', positive class prior π, $speed_n = K$
Output: Updated P', N' and model M_2

$P' = P$
$\delta^n = K$, $\delta^p = K \times \pi/(1-\pi)$
model $M = Classifier(P, N)$
$score = predict(M, U')$
for $t = 1 : (r\% \times (1-\pi) \times \mid U \mid - \mid N' \mid)/K + 1$ **do**
  $S = rank(score)$ in ascending order
  if $t < 1 : (r\% \times (1-\pi) \times \mid U \mid - \mid N' \mid)/K + 1$ **then**
    $N_{add} = \{x_i | (score(i) < S(K))\}$
    $P_{add} = \{x_i | (score(i) > S(end - K \times \pi/(1-\pi) + 1))\}$
  else
    $N_{add} = \{x_i | (score(i) < 0)\}$
    $P_{add} = \{x_i | (score(i) > 0)\}$
  end if
  $N' = N' \cup N_{add}$, $P' = P' \cup P_{add}$, $U' = U' - N_{add} - P_{add}$
  model $M_2 = Classifier(P', N')$
  $score = predict(M_2, U')$
end for

2) Self-learning with estimated positive class prior: After the pre-processing by the linear model, we have instances from both classes and remaining unlabeled set U'. An initial classifier trained by provided positive set P and extracted reliable negative set N' can be generated. Though the remaining unlabeled set can be divided by the initial classifier, there will be many false detections since the distribution of the two classes may overlap. To get a more accurate final boundary between two classes, self-learning process is used to keep adjusting the classifier by gradually adding reliable positive and negative instances into two classes.

The goal of the self-learning process is to gradually update the positive set P' and negative set N' by finding the most reliable negative and positive instances of the current state from the unlabeled set. The initial classifier built by the positive set P' and the negative set N' is applied to the remaining unlabeled set U'. The scores for all the unlabeled instances are ranked in ascending order. Those with the highest scores and lowest scores have the highest and lowest probability of being positive. They are the most reliable positive instances P_{add} and negative instances N_{add} which will be added to the two classes. Then a new classifier is built based on the updated set P' and N' to get new reliable instances. Along with the learning process, the positive and negative sets are getting more complete.

Self-learning speeds for positive and negative classes are different based on the estimated positive class prior π. The ratio of positive and negative instances numbers is $\pi/(1-\pi)$. Suppose the learning speed for the negative class is K, the learning speed for positive class will be $K \times \pi/(1-\pi)$. Added false positive and False negative will transfer the wrong information along the self-learning and have higher influence on later steps. To make sure the added instances are pure, the learning speed cannot be large. At the end of the self-learning step, $r\%$ of the unlabeled set are divided based on the estimated positive class prior. The remaining $(1-r\%)$ of the unlabeled set U are kept for the last step to correct the error in positive class prior estimation. The selection of ratio $r\%$ is derived in the next section.

3) Correction of error in positive class prior estimation: The positive class prior π is an estimated term. In the self-learning process, we add the reliable positive and negative instances according to this estimated positive class prior, which will lead to changing of positive class prior of remaining unlabeled data. Hence, the remaining unlabeled instances in the last step of self-learning process have different positive and negative ratios with either real or estimated positive class prior π. If we keep classifying remaining unlabeled instances based on the estimated positive class prior in the last step of self-learning, it will introduce error to the building of final positive set P' and negative set N'. To cancel the error induced by positive class prior estimation, we use a different way to classify the last step U' of the self-learning process. The remaining $(1-r\%)$ of the unlabeled set U will be divided based on their predicted labels. After the final step, all the unlabeled instances in training set are divided into positive or negative class. All the training data are utilized to train the final classifier and no training samples will be discarded in the training process.

The relationship between the error of estimated positive class prior and the proper $r\%$ is derived as follows. Suppose the real positive class prior is π_0 and the error of estimated positive class prior is $e = \frac{\pi - \pi_0}{\pi_0}$. The worst case that we want to avoid is the remaining $(1-r\%) \times size(U)$ instances all belong to one class, either positive or negative. When $e < 0$, the estimated positive class prior is smaller than the real positive class prior, i.e., we under-estimated the portion of positive instances. The worst situation for this case is that the remaining instances all belong to the positive class. Similarly, when $e > 0$, the worst situation is that the remaining instances all belong to the negative class. To avoid both situations, the constraints can be represented by:

$$r\% < \begin{cases} \frac{1-\pi}{1-\pi(e+1)} & \text{when } e < 0, \\ \frac{1}{1+e} & \text{when } e > 0. \end{cases} \tag{10}$$

It can also be shown that $\frac{1-\pi}{1-\pi(e+1)} < \frac{1}{1+e}$ always holds when $\pi > 0.5$. The smaller $r\%$ is picked to avoid both situations, thus the relationship between the error of estimated positive class prior and positive class prior $r\%$ that needs to be satisfied is:

$$r\% < \begin{cases} \frac{1-\pi}{1-\pi(e+1)} & \pi > 0.5, \\ \frac{1}{1+e} & \pi \leq 0.5. \end{cases} \tag{11}$$

In the anomaly detection scenario, the positive class prior is much larger than 0.5. The algorithm for the complete self-

learning process of proposed PU learning framework is shown in Algorithm 1 and the pipeline is shown in Fig. 2.

B. Boosting in self-learning process when feedback is available

Algorithm 2 Self-learning process with boosting when feedback is available

Input: Training set: P', U', N', positive class prior π, $speed_n = K$, $\%p$
Output: Updated P', N' and model C_{end}

$\delta^n = K$, $\delta^p = K \times \pi/(1-\pi)$
$\omega_1 \leftarrow ones(\mid U \mid, 1)$
for $t = 1 : (r\% \times (1-\pi) \times \mid U \mid - \mid N' \mid)/K + 2$ **do**
  $c_t = Classifier(P', N')$
  give feedback to random $\%p$ of N_{add}, P_{add} to get estimated error rate
  $\alpha_t = \frac{1}{2} ln(\frac{\sum_{y_i = c_t(x_i)} \omega_i^{(t)}}{\sum_{y_i \neq c_t(x_i)} \omega_i^{(t)}})$
  update ω_{t+1} as Eq.(18) for all x_i based on estimated error rate
  $C_t = \alpha_1 c_1 + ... + \alpha_{t-1} c_{t-1} + \alpha_t c_t$
  $score = predict(C_t, U')$
  if $t < (r\% \times (1-\pi) \times \mid U \mid - \mid N' \mid)/K + 1$ **then**
    $S = rank(score)$ in ascending order
    $N_{add} = \{x_i | (score(i) < S(K))\}$
    $P_{add} = \{x_i | (score(i) > S(end - K \times \pi/(1-\pi) + 1))\}$
  else if $t = (r\% \times (1-\pi) \times \mid U \mid - \mid N' \mid)/K + 1$ **then**
    $N_{add} = \{x_i | (score(i) < 0)\}$
    $P_{add} = \{x_i | (score(i) > 0)\}$
  else
    break
  end if
  $N' = N' \cup N_{add}$, $P' = P' \cup P_{add}$, $U' = U' - N_{add} - P_{add}$
end for

In the previous section, we introduce a complete structure of PU learning process and a final classifier can be built by training data. However, due to the lack of labels for unlabeled set U, the performance may not always improve along the process of self-learning. Along with more false positive and false negative instances added into set P' and N', the performance may stop improving and start to decrease at some stage. What we can do for the proposed structure in Section III (A) is to decrease the number of added instances to ensure they are reliable. However, when speed K is decreased, the number of self-learning steps will increases. The increase of steps will also lead to high chance of a performance decrease in later steps of self-learning. The number of self-learning steps and size of self-learning steps are balanced to get optimal performance. Nevertheless, the effect is not guaranteed.

To tackle this instability characteristic of the structure, we propose to apply boosting in the process of self-learning. Boosting is an ensemble machine learning method that can convert a set of weak learners to a strong learner. Boosting generally improves the performance of individual classifiers especially when the classifier is highly sensitive to small perturbation of the training set. Each classifier from self-learning process is treated as a weak individual classifier here. Boosting is adopted by weighted combining the series of classifiers. The proof of improvement of performance is explained as follows.

The classifier from each self-learning is reserved as c_t, the boosted classifier is a linear combination of all the classifiers up to this step:

$$\begin{aligned} C_t(x_i) &= \alpha_1 c_1(x_i) + ... + \alpha_{t-1} c_{t-1}(x_i) + \alpha_t c_t(x_i) \\ &= C_{(t-1)}(x_i) + \alpha_t c_t(x_i). \end{aligned} \tag{12}$$

The total error E of boosted classifier C_t is defined as the sum of exponential loss on each instance:

$$E = \sum_{i=1}^{N} e^{-y_i C_t(x_i)}. \tag{13}$$

We use ω_i to denote the exponential loss on instance x_i, given as follows:

$$\omega_i^t = \begin{cases} 1 & t = 1, \\ e^{-y_i C_{t-1}(x_i)} = \omega_i^{t-1} e^{-y_i \alpha_{t-1} c_{t-1}(x_i)} & t > 1. \end{cases} \tag{14}$$

Substitute Eq.(14) into Eq.(13), we have

$$\begin{aligned} E &= \sum_{i=1}^{N} \omega_i^{(t)} e^{-y_i \alpha_t c_t(x_i)} \\ &= \sum_{i=1}^{N} \omega_i^{(t)} e^{-\alpha_t} + \sum_{y_i \neq c_t(x_i)} \omega_i^{(t)} (e^{\alpha_t} - e^{-\alpha_t}). \end{aligned} \tag{15}$$

The weights $(\alpha_t)_{t=1}^T$ are determined to get better boosted classifier C_t comparing to $C_{(t-1)}$. It can be achieved if E has the lowest value when α_t is selected. Thus we will minimize E with new classifier c_t, which is to set

$$\frac{dE}{d\alpha_t} = \sum_{y_i \neq c_t(x_i)} \omega_i^{(t)} e^{\alpha_t} - \sum_{y_i = c_t(x_i)} \omega_i^{(t)} e^{-\alpha_t} = 0. \tag{16}$$

It gives α_t as

$$\alpha_t = \frac{1}{2} ln(\frac{\sum_{y_i = c_t(x_i)} \omega_i^{(t)}}{\sum_{y_i \neq c_t(x_i)} \omega_i^{(t)}}). \tag{17}$$

Thus, the weight α_t can be obtained from current step ω_i^t. From Eq.(14) we can tell that ω_i^t is related to α_{t-1} and the prediction correctness $y_i \times c_{t-1}(x_i)$. We can update Eq.(14) to get ω_i^t by prediction result of $C_{t-1}(x_i)$:

$$\omega_i^t = \begin{cases} \omega_i^{t-1} e^{-\alpha_{t-1}} & y_i = c_{t-1}(x_i), \\ \omega_i^{t-1} e^{\alpha_{t-1}} & y_i \neq c_{t-1}(x_i). \end{cases} \tag{18}$$

To get the weight α_t for each classifier, we need to first get ω^t which can only be measured when labels are provided. Though we cannot get the true labels y_i for each instance x_i in set U, for some real-world applications, we can have feedback for a portion of U in self-learning process when the budget is available. Thus we can measure the error rate by randomly selecting $\%p \times K$ of the remaining unlabeled set and give them the feedback at each step of self-learning. Here $\%p$ is a small ratio that will be selected and we use 20% for

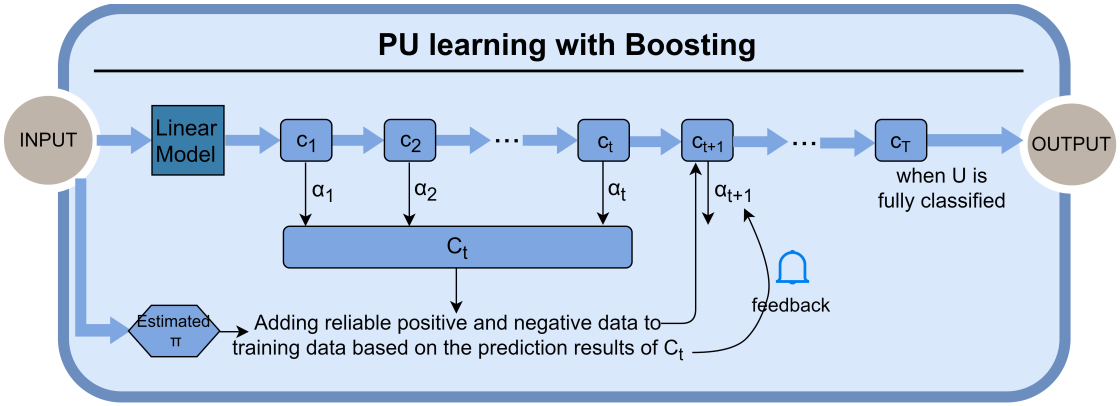


Fig. 3: Architecture for proposed boosting PU learning method. The boosted classifier is a weighted sum of weak classifiers from each self-learning step. The weight α_t for each classifier is computed based on the performance of the current stage. π in the graph denoted the positive class prior.

all datasets in the experiment. Those candidates will be added to the positive set P' and negative set N' accordingly with the real labels. Then error rate er of the current step classifier c_t on selected instances is calculated.

Since ω is a vector of the same length with the unlabeled set and it is updated according to Eq.(18), we randomly assign new ω_i^t to the remaining unlabeled set according to the estimated error rate. Finally the weight α_t is updated as Eq.(17). Thus, we have the updated boosted classifier for the prediction of remaining unlabeled set. The scores for the instances are sorted to select the most reliable instances for the two classes. The last step of self-learning still needs to classify based on the predicted label instead of the relative speed decided by estimated positive class prior to narrow the error of estimated positive class prior.

The boosting structure in PU learning is like a memory device. The weights decide how much we should memorize from each classifier alone the self-learning process. The prediction of remaining unlabeled set is based on the weighted combination of these classifiers. The weight α_t of each memory cell depends on the error rate of the current step classifier. The algorithm for the self-learning process with boosting when feedback is available is shown in Algorithm 2 and how to construct the combined classifier C_t for step t is illustrated in Fig. 3. Although the proof is based on the true error rate of each classifier and our method is based on the estimated error rate, experimental results on six datasets validate the proposed boosting PU learning method will improve the performance.

IV. Experiment

A. Evaluation Measures

Following the evaluation of classification task, we use F score as the measurement. F score is obtained from precision and recall, whose formulas are listed as:

$$Precision = \frac{TP}{(TP + FP)} = \frac{TP^*}{\text{all detected anomalies}}$$

$$Recall = \frac{TP}{(TP + FN)} = \frac{TP^*}{\text{all real anomalies}}$$

$$Fscore = \frac{2 \times Precision \times Recall}{(Precision + Recall)}$$

where TP, FP and FN stand for true positive, false positive and false negative correspondingly. The P^* in the evaluation matrix is different from the positive class in the training process. It denotes the detection target, which is the anomaly class.

B. Experiment Dataset and Settings

The evaluated datasets are from diverse fields of anomaly detection: Cyber security (Kyoto dataset); Image dataset (MNIST dataset Statlog (Landsat Satellite) dataset); Video dataset (UMN dataset); Safety (Statlog (Shuttle) dataset); and also a synthetic dataset.

- **Kyoto dataset**: Kyoto dataset is a widely used performance evaluation dataset in the intrusion detection research field. It contains 24 statistical features: 14 conventional features are significant and essential features extracted from the raw traffic data (KDD Cup 99 dataset) obtained by honeypot systems of Kyoto University; another 10 additional features are extracted to investigate more effectively what happens on the networks. We use 20 out of the 24 features for the experiment and randomly select 100k instance as training set.
- **MNIST dataset**: MNIST dataset (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. Principal component analysis (PCA) is applied to the original features to get a 2-dimensional feature for the later training process. We randomly choose 6000 images for training.
- **UMN dataset**: UMN dataset is a video dataset for detecting unusual crowd activity such as crowds running in one direction, or a crowd of people dispersing from a central point. The video is split into 16 frames long clips with a 15-frame overlap between two consecutive clips. Each clip is labeled as normal or abnormal. The spatiotemporal features of 64×64 dimensions for each

TABLE I: EXPERIMENTAL DATASETS

Dataset	#Features	#classes	positive class prior	#training instances	positive set % in training set
Kyoto dataset	20	2	22%	100,000	20%,40%,60%,80%
MNIST dataset	784	10	10%,20%,...,90%	6000	50%
UMN dataset	64×64	2	90%	3000	30%,40%,50%,60%
Statlog (Shuttle) dataset	9	2	73%	32732	15%
Statlog (Landsat Satellite) dataset	36	2	63%	3717	27%
Synthetic dataset	2	2	90%	600	30%

video clip are learnt with 3d convolutional network [34]. 3000 video clips are randomly selected for training.

- **Statlog (Shuttle) dataset**: Statlog (Shuttle) dataset is a multi-class classification dataset with 9 numerical attributes. The smallest five classes, i.e. 2, 3, 5, 6, 7 are combined to form the outliers class, while class 1 forms the inlier class. Data from class 4 are discarded.
- **Statlog (Landsat Satellite) dataset**: Satellite dataset is a multi-class classification dataset, which consists of the multi-spectral values of pixels in 3x3 neighborhoods in a satellite image. There are 7 classes and the smallest three classes, i.e. 2, 4, 5 are combined to form the outlier class, while all the other classes are combined to form an inlier class.
- **Synthetic dataset**: A synthetic dataset is generated to vary the distribution overlapping of positive and negative classes. We use Gaussian distribution to simulate 2-dimensional data with distribution characters of anomaly detection datasets. The normal data are more dense cluster while the anomalies are multi-clusters with larger variation. When the distribution overlapping of two classes is larger, the boundary of two classes is more difficult to identify. More anomalies behave similarly to normal data and more learning should focus on the confusing region. 600 instances are randomly selected for each training set.

Other details of the datasets are listed in Table I.

For all the datasets in experiments, we only utilize labeled positive data and unlabeled data for training and mixed data for testing. The structure of the training data is illustrated in Fig. 4. As shown in the figure,

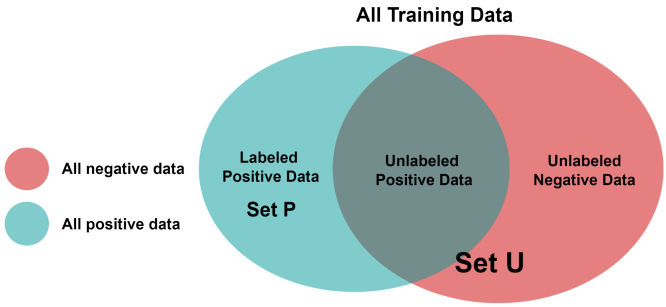


Fig. 4: Illustration of data structure and validation perspective.

$$\%\ of\ P = \frac{P}{P \cup U}$$

$$class\ prior\ \pi = \frac{Positive\ data\ in\ U}{U}$$

C. Existing Methods for Comparison

- One-class SVM [35]: Only the labeled positive set P is utilized to build the one-class SVM model. It cannot make use of the information in provided unlabeled data.
- S-EM [5]: It is a widely used PU learning method which uses spy-positive instances to identify most likely negative instances from the unlabeled data. Then Expectationmaximization (EM) algorithm is employed to obtain the final classifier.
- LPU [36], [6]: It is a widely used PU learning method for text learning. It first regards all the unlabeled data as negative and utilizes Rocchio classifier to identify the reliable negative instances. Then EM algorithm is iteratively applied to instances that are more reliable and the good classifier is caught during the process.
- Convex Linear model [13]: It is a Linear-in-parameter model for positive and unlabeled problem. It proposes to use different loss functions for positive and unlabeled set. The loss function for unlabeled set is chosen as squared loss function $l_s(z) = \frac{1}{4}(z-1)^2$, which is the same as our pre-training process (Section III (A) 1)).
- Same classifier with fully labeled training data (denoted by PN): With this setting, all labels for original unlabeled set U will be given, which means we have both labeled negative and positive data. The classifier is selected as Random Forest, which is consistent with proposed methods for comparison. We use this setting to validate whether the proposed methods can have comparable performance with lack of negative class information.

We use Ours1 and Ours2 to represent the proposed PU learning framework and boosting PU learning method respectively.

D. Experiment Results

The F scores for the testing set of the six datasets are shown in Table II. The experimental settings are shown as Table I and the special selection for each dataset is indicated in Table II. The performance of OCSVM, S-EM, LPU, Convex linear model and PN is compared with the proposed PU learning method and boosting PU learning method. We can have an overview from Table II that the proposed methods improve the performance for different kinds of datasets. The proposed boosting PU learning method achieves state-of-the-art performance. The bagging SVM method [32] is also tried, but it cannot detect any anomalies in all the datasets since the unlabeled set is highly contaminated by positive instances. The experimental results for all datasets are average performance of 10 randomly selected testing sets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2871421, IEEE Transactions on Multimedia

TABLE II: F SCORE OF ALL DATASETS

F score	OCSVM	S-EM [5]	LPU [36], [6]	Convex Linear model [13]	Ours1	Ours2	PN
Kyoto dataset (% of P = 40%)	75.70%	59.26%	81.85%	87.69%	91.79%	94.86%	98.64%
MNIST dataset (π = 0.9)	66.05%	65.00%	0%	64.60%	79.22%	80.54%	72.18%
UMN dataset (% of P = 30%)	28.62%	26.01%	26.01%	92.55%	93.43%	94.46%	97.75%
Statlog (Shuttle) dataset	57.82%	27.76%	0%	65.20%	73.92%	82.91%	99.07%
Statlog (Landsat Satellite) dataset	58.86%	45.92%	0%	6.71%	32.90%	62.98%	89.54%
synthetic dataset (distribution overlapping ratio=25%)	28.54%	32.34%	25.96%	24.63%	60.04%	63.31%	64.23%

TABLE III: POSITIVE CLASS PROPORTION VALIDATION (KYOTO DATASET)

% of P	OCSVM			S-EM [5]			LPU [36], [6]			Convex Linear model [13]		
	Precision	Recall	F score	Precision	Recall	F score	Precision	Recall	F score	Precision	Recall	F score
20%	68.05%	99.99%	**80.13%**	70.34%	64.15%	**58.21%**	89.70%	82.17%	**84.43%**	95.07%	84.88%	**89.46%**
40%	62.53%	100.00%	**75.70%**	79.95%	62.07%	**59.26%**	88.50%	80.98%	**81.85%**	91.20%	85.25%	**87.69%**
60%	68.04%	100.00%	**80.21%**	74.88%	67.40%	**58.61%**	90.48%	77.43%	**81.22%**	92.78%	84.28%	**88.13%**
80%	64.92%	99.99%	**77.62%**	72.71%	83.15%	**73.49%**	88.38%	84.90%	**85.47%**	88.76%	84.70%	**85.95%**

% of P	Ours1			Ours2			PN		
	Precision	Recall	F score	Precision	Recall	F score	Precision	Recall	F score
20%	99.05%	86.37%	**92.17%**	99.07%	90.06%	**94.34%**	97.76%	99.88%	**98.81%**
40%	98.57%	86.09%	**91.79%**	98.02%	92.10%	**94.86%**	97.70%	99.64%	**98.65%**
60%	99.13%	86.46%	**92.29%**	98.96%	90.75%	**94.64%**	98.59%	98.06%	**98.30%**
80%	99.45%	81.42%	**89.14%**	99.35%	90.23%	**94.52%**	99.36%	92.88%	**95.99%**

TABLE IV: POSITIVE CLASS PROPORTION VALIDATION (UMN DATASET)

% of P	OCSVM			S-EM [5]			LPU [36], [6]			Convex Linear model [13]		
	Precision	Recall	F score	Precision	Recall	F score	Precision	Recall	F score	Precision	Recall	F score
30%	16.70%	100%	**28.62%**	14.95%	100.00%	**26.01%**	14.95%	100.00%	**26.01%**	95.75%	89.78%	**92.55%**
40%	16.93%	100%	**28.96%**	14.57%	100.00%	**25.43%**	14.57%	100.00%	**25.43%**	96.06%	81.46%	**86.69%**
50%	17.75%	100%	**30.15%**	14.82%	100.00%	**25.81%**	14.82%	100.00%	**25.81%**	97.51%	80.43%	**86.96%**
60%	18.15%	100%	**30.72%**	14.61%	100.00%	**25.49%**	14.61%	100.00%	**25.49%**	96.06%	89.04%	**92.30%**

% of P	Ours1			Ours2			PN		
	Precision	Recall	F score	Precision	Recall	F score	Precision	Recall	F score
30%	93.73%	93.43%	**93.43%**	94.61%	94.47%	**94.46%**	98.39%	96.69%	**97.75%**
40%	93.13%	92.44%	**92.24%**	92.28%	94.28%	**92.98%**	98.82%	95.38%	**97.07%**
50%	94.84%	95.28%	**95.02%**	94.58%	94.48%	**94.38%**	99.00%	94.65%	**96.77%**
60%	95.74%	93.36%	**94.50%**	95.48%	93.62%	**94.46%**	99.11%	93.76%	**96.35%**

1) Positive class proportion validation: The experiment is conducted to validate that the proposed methods are not sensitive to different positive set P proportion. The Kyoto dataset and UMN dataset are tested under different P proportions. For Kyoto dataset, the positive class prior is similar to original raw dataset and percentage of labeled positive set P is varied as 20%, 40%, 60% and 80%. For UMN dataset, the positive class prior is set as 90% and the percentage is varied from 30% to 60%.

The performance of OCSVM, S-EM, LPU, Convex linear model and PN is also compared with the proposed methods. Average experimental results of 10 sets are shown in Table III and Table IV. We can tell that the performance will not be influenced significantly when the proportion of positive set changes. The proposed methods outperform all the other methods with the same training settings under different portions of positive set P in the training set. The performance approaches the PN method whose labels for training data are all provided. From the parameters of linear model and random forest, we could identify the features that are more discriminative. The strong predictors of Kyoto dataset of multi-features setting are: session start time, percentage of connection that 'SYN' errors in Dst_host_srv_count, the connection's service type and percentage of connection that have 'SYN' errors in Dst_host_count [37].

2) Positive class prior distribution validation: The experiment is conducted to validate that the proposed methods can work well not only in anomaly detection scenario but also in other data distributions. The MNIST dataset is re-formulated with different positive class priors π from 0.1 to 0.9. The proportion of positive set P of MNIST dataset is set at 50%.

The performance of OCSVM, S-EM, Convex linear model and PN is also compared with the proposed methods. Average results of 10 sets are shown in Table V. The performance

TABLE V: PRIOR DISTRIBUTION VALIDATION (MNIST DATASET)

positive class prior π	OCSVM			S-EM [5]			Convex Linear model [13]		
	Precision	Recall	F score	Precision	Recall	F score	Precision	Recall	F score
0.1	98.89%	75.92%	**83.92%**	86.79%	59.25%	**68.27%**	95.46%	98.94%	**97.10%**
0.2	98.15%	77.47%	**84.69%**	87.14%	58.53%	**67.67%**	90.50%	99.35%	**94.52%**
0.3	95.39%	75.28%	**82.21%**	88.37%	57.73%	**67.16%**	82.74%	99.57%	**90.00%**
0.4	94.80%	77.93%	**83.92%**	87.88%	56.94%	**66.57%**	74.51%	99.40%	**84.96%**
0.5	92.48%	73.06%	**79.67%**	87.24%	58.09%	**67.65%**	63.70%	99.33%	**77.06%**
0.6	90.67%	77.27%	**82.04%**	86.59%	58.15%	**68.08%**	67.02%	98.32%	**78.92%**
0.7	86.09%	73.54%	**78.04%**	83.76%	62.08%	**69.50%**	63.87%	97.09%	**75.39%**
0.8	77.20%	77.33%	**76.68%**	84.21%	59.90%	**68.79%**	62.65%	96.89%	**74.59%**
0.9	60.88%	74.75%	**66.05%**	76.81%	58.38%	**65.00%**	52.05%	90.91%	**64.60%**
positive class prior π	Ours1			Ours2			PN		
	Precision	Recall	F score	Precision	Recall	F score	Precision	Recall	F score
0.1	98.68%	91.21%	**94.74%**	98.73%	91.79%	**95.05%**	98.22%	93.63%	**95.82%**
0.2	97.91%	91.05%	**94.32%**	97.88%	91.44%	**94.48%**	96.97%	93.55%	**95.20%**
0.3	95.81%	91.05%	**93.34%**	95.79%	91.15%	**93.38%**	94.35%	91.73%	**92.99%**
0.4	94.72%	91.50%	**92.99%**	95.22%	91.13%	**93.04%**	93.16%	91.65%	**92.38%**
0.5	92.91%	90.24%	**91.53%**	93.38%	88.78%	**90.95%**	90.96%	90.35%	**90.63%**
0.6	92.99%	89.73%	**91.27%**	92.49%	90.40%	**91.40%**	89.07%	90.24%	**89.53%**
0.7	88.03%	86.51%	**87.11%**	88.81%	86.77%	**87.71%**	85.68%	85.71%	**85.50%**
0.8	88.60%	87.11%	**87.77%**	88.04%	87.56%	**87.70%**	84.75%	86.22%	**85.41%**
0.9	89.41%	74.74%	**79.22%**	86.11%	76.77%	**80.54%**	71.54%	73.74%	**72.18%**

of all listed methods will decrease when positive class prior increases. This is because when positive data portion in unlabeled data increases, the unlabeled set will be higher contaminated by the positive instances. The unlabeled set U is harder to be differentiated with the positive set P. Also, the estimated error of positive class prior will have a larger influence on the self-learning relative speed. From Table V, it is clear that the proposed methods outperform all the other methods with the same training settings. The performance is even better than the PN method whose labels for training data are all provided. The performance improvement is larger when positive class prior is very large, i.e. in anomaly detection scenario.

3) Distribution overlapping ratio validation: The experiment is conducted to validate that the proposed methods have better performance compared to existing methods when the two classes have different overlapping ratios. To achieve the experimental settings that can decide the different overlapping ratios of two distributions, we synthesize a dataset with 2-dimensional Gaussian distribution, which is easy to model the distribution overlaps. The synthetic dataset is simulated according to the anomaly detection scenario: the normal instances belong to a large and dense cluster, while anomalies belong to small or sparse multi-clusters. Thus, the variations of anomalies are much larger than normal instances. The positive class prior is set at 90% and the positive class proportion is set at 30% in the experiment. The overlapping ratios of the two classes are varied from 10% to 45%.

The performance of OCSVM, S-EM, LPU, Convex linear model and PN is compared with the proposed methods and the average performance is shown in Fig. 5. Larger overlapping ratios of two classes distributions signify how likely anomalies behave like normal instances. As we predicted, the performance of all listed methods will decrease while overlapping ratio increases. The curves in Fig. 5 also indicate that the proposed methods outperform all the other methods with the same training settings. The F score value is approaching to the PN performance and sometimes even performs better than PN method. A set of experiment of 15% distribution overlapping results is shown in 2D plot explicitly as Fig. 6. We can see from the circles differences, proposed boosting PU learning (e) has better performance near the separation boundary of the two classes comparing with proposed PU learning method (d).

TABLE VI: SELF-LEARNING STEP SIZE VALIDATION (KYOTO DATASET (% of P = 40%))

K	Ours1			Ours2		
	Precision	Recall	F score	Precision	Recall	F score
800	98.28%	85.59%	**91.42%**	98.99%	88.58%	**93.41%**
1000	98.57%	86.09%	**91.79%**	98.02%	92.10%	**94.86%**
1200	98.63%	85.87%	**91.73%**	98.69%	88.73%	**93.37%**
1500	98.57%	86.93%	**92.30%**	98.98%	88.00%	**93.11%**
1700	98.81%	85.93%	**91.86%**	98.68%	88.89%	**93.43%**
2000	98.84%	85.34%	**91.47%**	98.67%	88.58%	**93.29%**

4) Parameters selection: We have conducted experiments on Kyoto dataset to validate the effect of self-learning step size K for proposed methods. The proportion of positive set P of Kyoto dataset is set at 40%. The previous experiment results of Kyoto dataset are conducted with step size K of 1000. The step size for this experiment is varied from 800 to 1700. Average results of 10 sets from proposed methods are shown in Table VI. The performance is the best when step size is 1500 for proposed vanilla PU learning method and 1000 for proposed boosting PU leaning method. A appropriate value should be selected for the self-learning step size K. This is because when step size K is too large, the added positive and negative instances are not reliable enough. When step size is too small, we need more self-learning steps to classify the unlabeled data which will also lead to high contaminate in the updated set P' and N'. It can be shown from the table that our performance is not very sensitive to step size K and the performance is guaranteed. From the Table, it is clear

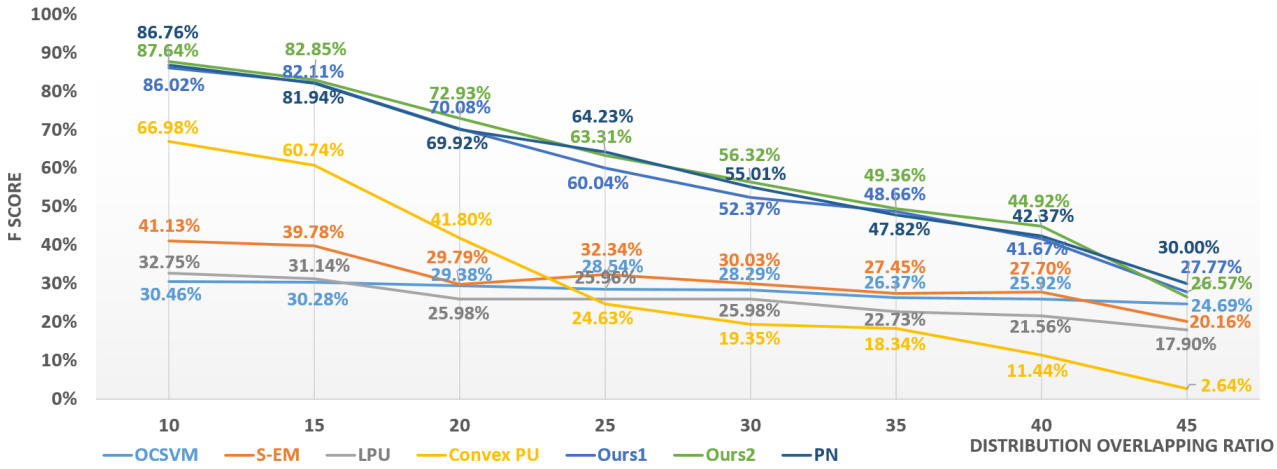


Fig. 5: Performance for different ratios of distribution overlapping for two classes.

that the proposed boosting PU learning methods have better performance than the proposed vanilla PU learning method. The reason is that boosting improves the stability of the self-learning process.

TABLE VII: PRE-TRAINING NEGATIVE SAMPLE RATIO VALIDATION (KYOTO DATASET (% of P = 40%))

s%	Ours1			Ours2		
	Precision	Recall	F score	Precision	Recall	F score
80%	98.79%	85.33%	**91.45%**	98.64%	88.26%	**93.08%**
70%	98.57%	86.09%	**91.79%**	98.02%	92.10%	**94.86%**
60%	99.22%	84.30%	**91.05%**	99.03%	88.26%	**93.27%**
50%	99.49%	77.32%	**86.53%**	99.20%	88.30%	**93.36%**

Experiments to validate the effect of number of negative samples we extracted during the pre-training of proposed methods is also conducted on Kyoto dataset. The proportion of positive set P of Kyoto dataset is set at 40% and step size K is set as 1000. For the previous experiment results of Kyoto dataset, we extracted 70% of the estimated number of negative samples. In this validation, we varied s% from 40% to 80%. Average results of 10 sets from proposed methods are shown in Table VII. From the Table, we can tell the performance is the best when we select the ratio s% as 70%. We could not extract the exact estimated number of negative samples due to two facts. The first fact is the accuracy of the pre-training linear model is not 100%. Only the samples behave quite differently from the results of the pre-training model have the highest accuracy to be truth anomalies. The other reason is that the class prior is an estimation value. Thus, an appropriate value of s% is selected to make the extracted negative samples reliable and the amount of negative samples big enough to train initial two-class model in the self-learning process.

We have conducted another experiments on MNIST dataset to analysis performance with different dimension d from PCA. The class prior π of MNIST dataset is set at 0.6. In this validation, dimension d kept from PCA for MNIST dataset is varied from 2 to 784. Average results of 10 sets from proposed methods are shown in Table VIII. From the Table, we can tell

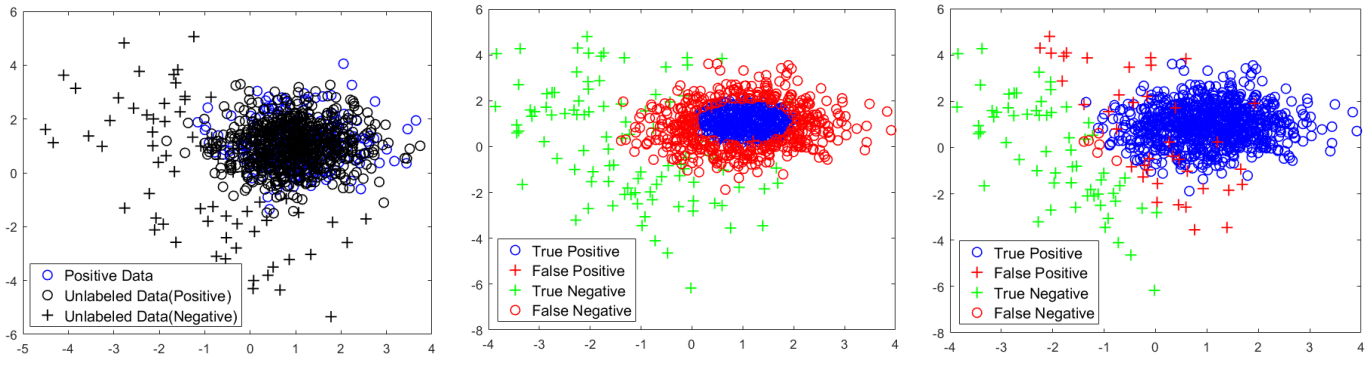
TABLE VIII: DIMENSION FROM PCA VALIDATION (MNIST DATASET (π = 0.6))

d	Ours1			Ours2		
	Precision	Recall	F score	Precision	Recall	F score
2	92.99%	89.73%	**91.27%**	92.49%	90.40%	**91.40%**
5	98.04%	94.11%	**95.94%**	97.93%	94.62%	**96.19%**
10	95.55%	93.43%	**94.38%**	97.00%	92.93%	**94.73%**
20	93.66%	88.89%	**90.69%**	94.37%	88.89%	**91.21%**
50	92.37%	80.47%	**85.28%**	92.70%	81.82%	**86.43%**

the performance is the best when we select the dimension d as 10. The dimension of feature space from PCA is not the higher the better for the training. The redundant information in high dimension may even affect the classification. The MNIST dataset is a database with simple handwritten digits, making the required dimension of feature space from PCA not high. We still show the remaining experiment with 2-dimensional features from PCA because we compare our results with linear model [13] which uses the same setting of maintain only 2-dimensional features from PCA.

V. Conclusion

In this paper, we propose two effective Positive and Unlabeled (PU) learning frameworks on anomaly detection problem. We apply PU learning method to the anomaly detection problem with multi-features to avoid the labeling of various types of anomalies. To tackle unbalanced data distribution where unlabeled set is dominated by the normal data, a new framework is proposed to first identify the reliable negative instances followed by a self-learning process based on the estimated positive class prior. The boosting process is then incorporated in the self-learning process to get appropriate weights for each classifier of the self-learning process to tackle the instability of PU learning process. Extensive experiments are conducted on seven datasets under different settings and achieve state-of-the-art performance under the same training data settings.



(a) Data Distribution of training data (b) One-class SVM (c) Convex Linear model [13]

(d) Proposed PU learning method (e) Proposed boosting PU learning method (f) PN: training with all negative labels given

Fig. 6: Detection results comparison for 15% distribution overlapping for two classes. x, y axes represent the feature vectors and z axis represents the probability density of the location.

REFERENCES

[1] M. Saini, X. Wang, P. K. Atrey, and M. Kankanhalli, "Adaptive workload equalization in multi-camera surveillance systems," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 555–562, 2012.

[2] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "Probabilistic novelty detection for acoustic surveillance under real-world conditions," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 713–719, 2011.

[3] T. Chen, L.-A. Tang, Y. Sun, Z. Chen, and K. Zhang, "Entity embedding-based anomaly detection for heterogeneous categorical events," *arXiv preprint arXiv:1608.07502*, 2016.

[4] Y.-Y. Chen, W. H. Hsu, and H.-Y. M. Liao, "Automatic training image acquisition and effective feature selection from community-contributed photos for facial attribute detection," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1388–1399, 2013.

[5] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *ICML*, vol. 2. Citeseer, 2002, pp. 387–394.

[6] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *IJCAI*, vol. 3, no. 2003, 2003, pp. 587–592.

[7] H. Li, B. Liu, A. Mukherjee, and J. Shao, "Spotting fake reviews using positive-unlabeled learning," *Computación y Sistemas*, vol. 18, no. 3, pp. 467–475, 2014.

[8] Y. Ren, D. Ji, and H. Zhang, "Positive unlabeled learning for deceptive reviews detection." in *EMNLP*, 2014, pp. 488–498.

[9] F. Mordelet and J.-P. Vert, "Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples," *BMC bioinformatics*, vol. 12, no. 1, p. 389, 2011.

[10] P. Yang, X.-L. Li, J.-P. Mei, C.-K. Kwoh, and S.-K. Ng, "Positive-unlabeled learning for disease gene identification," *Bioinformatics*, vol. 28, no. 20, pp. 2640–2647, 2012.

[11] P. Yang, X. Li, H.-N. Chua, C.-K. Kwoh, and S.-K. Ng, "Ensemble positive unlabeled learning for disease gene identification," *PloS one*, vol. 9, no. 5, p. e97079, 2014.

[12] M. C. du Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in *Advances in neural information processing systems*, 2014, pp. 703–711.

[13] M. Du Plessis, G. Niu, and M. Sugiyama, "Convex formulation for learning from positive and unlabeled data," in *International Conference on Machine Learning*, 2015, pp. 1386–1394.

[14] J. Zhang, Z. Wang, J. Yuan, and Y.-P. Tan, "Positive and unlabeled learning for anomaly detection with multi-features," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017.

[15] S. Zhou, W. Shen, D. Zeng, M. Fang, Y. Wei, and Z. Zhang, "Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Signal Processing: Image Communication*, vol. 47, pp. 358–368, 2016.

[16] L. Xiao, Y. Chen, and C. K. Chang, "Bayesian model averaging of bayesian network classifiers for intrusion detection," in *Computer Software and Applications Conference Workshops (COMPSACW), 2014 IEEE 38th International*. IEEE, 2014, pp. 128–133.

[17] W. Li, G. Wu, and Q. Du, "Transferred deep learning for anomaly detection in hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 597–601, 2017.

[18] M. Yang, S. Rajasegarar, A. S. Rao, C. Leckie, and M. Palaniswami, "Anomalous behavior detection in crowded scenes using clustering and spatio-temporal features," in *Intelligent Information Processing VIII: 9th IFIP TC 12 International Conference, IIP 2016, Melbourne, VIC, Australia, November 18-21, 2016, Proceedings 9*. Springer, 2016, pp. 132–141.

[19] A. Ghosh and P. Gudipati, "Anomaly detection in web graphs using vertex neighbourhood based signature similarity methods," in *Data Science and Engineering (ICDSE), 2016 International Conference on*. IEEE, 2016, pp. 1–6.

[20] S. Ando, "Clustering needles in a haystack: An information theoretic analysis of minority and outlier detection," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 13–22.

[21] A. Agovic, A. Banerjee, A. R. Ganguly, and V. Protopopescu, "Anomaly detection in transportation corridors using manifold embedding," *Knowledge Discovery from Sensor Data*, pp. 81–105, 2008.

[22] W. Hu, X. Ding, B. Li, J. Wang, Y. Gao, F. Wang, and S. Maybank, "Multi-perspective cost-sensitive context-aware multi-instance sparse

coding and its application to sensitive video recognition," *IEEE Transactions on Multimedia*, vol. 18, no. 1, pp. 76–89, 2016.

[23] C. Alippi, S. Ntalampiras, and M. Roveri, "An hmm-based change detection method for intelligent embedded sensors," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–7.

[24] I. Ahn and C. Kim, "Face and hair region labeling using semi-supervised spectral clustering-based multiple segmentations," *IEEE Transactions on Multimedia*, vol. 18, no. 7, pp. 1414–1421, 2016.

[25] M. Jian and C. Jung, "Semi-supervised bi-dictionary learning for image classification with smooth representation-based label propagation," *IEEE Transactions on Multimedia*, vol. 18, no. 3, pp. 458–473, 2016.

[26] S. Wu, Q. Ji, S. Wang, H.-S. Wong, Z. Yu, and Y. Xu, "Semi-supervised image classification with self-paced cross-task networks," *IEEE Transactions on Multimedia*, 2017.

[27] L. Zhang, B. Ma, G. Li, Q. Huang, and Q. Tian, "Generalized semi-supervised and structured subspace learning for cross-modal retrieval," *IEEE Transactions on Multimedia*, 2017.

[28] Y. Xiao, B. Liu, J. Yin, L. Cao, C. Zhang, and Z. Hao, "Similarity-based approach for positive and unlabeled learning," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, 2011, p. 1577.

[29] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, no. 2, pp. 103–134, 2000.

[30] R. Xia, X. Hu, J. Lu, J. Yang, C. Zong *et al.*, "Instance selection and instance weighting for cross-domain sentiment classification via pu learning." in *IJCAI*, 2013, pp. 2176–2182.

[31] D. H. Fusilier, M. Montes-y Gómez, P. Rosso, and R. G. Cabrera, "Detecting positive and negative deceptive opinions using pu-learning," *Information processing & management*, vol. 51, no. 4, pp. 433–443, 2015.

[32] F. Mordelet and J.-P. Vert, "A bagging svm to learn from positive and unlabeled examples," *Pattern Recognition Letters*, vol. 37, pp. 201–209, 2014.

[33] M. C. Du Plessis and M. Sugiyama, "Class prior estimation from positive and unlabeled data," *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 5, pp. 1358–1362, 2014.

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[35] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[36] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003, pp. 179–186.

[37] J. Song, H. Takakura, and Y. Okabe, "Cooperation of intelligent honeypots to detect unknown malicious codes," in *Information Security Threats Data Collection and Sharing, 2008. WISTDCS'08. WOMBAT Workshop on*. IEEE, 2008, pp. 31–39.