Positive and unlabeled learning for anomaly detection

Zhang, Jiaqi

2018

Zhang, J. (2018). Positive and unlabeled learning for anomaly detection. Master's thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/75883>

<https://doi.org/10.32657/10356/75883>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**POSITIVE AND UNLABELED LEARNING
FOR ANOMALY DETECTION**

ZHANG JIAQI

**SCHOOL OF ELECTRICAL & ELECTRONIC
ENGINEERING**

2018

POSITIVE AND UNLABELED LEARNING FOR ANOMALY DETECTION

ZHANG JIAQI

ZHANG JIAQI

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Master of Engineering

2018

Acknowledgment

I would like to express my deepest appreciation to all those who provided me help for my research work. Firstly, I would like to express my sincere thanks and great gratitude to my supervisor, Professor Junsong Yuan for his encouragement and patient instruction. He guided me into the attractive research realm of machine learning with his open-mindedness and conscientiousness. I would also like to express my special appreciation to my co-supervisor Professor Tan Yap-Peng for his great support and guidance through my study.

Besides, my gratefulness should also go to all of the excellent members in my research group. Their brilliant thoughts inspired me a lot on research and encourage me to face the challenges. Furthermore, I would like to thank my family and my friends for their company.

Last but not least, a sincere the School of Electrical & Electronic Engineering, Nanyang Technological University (NTU), for providing me with the nice environment and equipment for my research work.

Acknowledgment

Contents

Acknowledgment	i
Contents	iii
Summary	vii
List of abbreviations	ix
List of tables	xi
List of figures	xiii
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Objectives	3
1.4 Contributions	4
1.5 Organization of the Report	5
2 Literature Review	7
2.1 Anomaly Detection	7
2.2 Positive and Unlabeled Learning	10
2.2.1 Methods based on spy technique [1]	12

Contents

2.2.2	Similarity or clustering based methods	14
2.2.3	Linear classifier based methods	15
2.2.4	Probability based methods	16
2.2.5	Streaming based methods	16
3	Positive and Unlabeled Learning for Anomaly Detection	19
3.1	Introduction	19
3.2	Proposed Approach: PU Learning for Anomaly Detection	21
3.2.1	Pre-training using Linear Model	21
3.2.2	Self-learning with estimated positive class prior	24
3.2.3	Final Detector Building	25
3.3	Experiment	27
3.3.1	Evaluation Measures	27
3.3.2	Experiment Dataset and Settings	28
3.3.3	Existing Methods for Comparison	30
3.3.4	Experiment Results and Discussion	31
4	Boosting Positive and Unlabeled Learning for Anomaly Detection	37
4.1	Introduction	37
4.2	Boosting in self-learning process when feedback is available	38
4.3	Experiment	41
4.3.1	Experiment Dataset and Settings	41
4.3.2	Existing Methods for Comparison	43
4.3.3	Experiment Results and discussions	43
5	Conclusions and Future Work	47
5.1	Conclusions	47

Contents

5.2	Future Work	48
5.2.1	anomaly detection of heterogeneous categorical events	49
5.2.2	PU learning for anomaly detection by deep learning	49
	List of publications	52
	Bibliography	53

Contents

Summary

Anomaly detection is of great interest to big data applications, and both supervised and unsupervised learning methods have been applied for anomaly detection. However, it remains a challenging problem for machine learning-based methods because of the difficulty of obtaining anomaly data for training, which is usually rare, diversely distributed, and difficult to collect. Thus, it is difficult for supervised learning to acquire training data; while for unsupervised learning, the performance may not be satisfactory due to the lack of labeled training data.

To address the challenge, we propose a hybrid solution by applying Positive and Unlabeled (PU) Learning for anomaly detection problem. As a semi-supervised method, only normal (positive) data and unlabeled data (could be positive or negative) are required by the proposed method for anomaly detection. We start by using a novel linear model to extract the most reliable negative instances followed by an iterative self-learning process to add reliable negative and positive instances with different speeds based on the estimated positive class prior. Our proposed method is verified on several benchmark datasets and one synthetic dataset. Experimental results show that our method outperforms existing methods under different class priors and different proportions of given positive classes.

Furthermore, we adopt boosting in the self-learning process to advantageously exploit the instability characteristic of PU learning when feedback is available. The classifiers in self-learning process are weighted combined based on the estimated

Summary

---

error rate to build the final classifier. Theoretical proof is provided to validate the performance improvement of adding boosting. The experiments demonstrate the effectiveness of our approach under different conditions compared to existing methods.

List of Abbreviations

BMA	Bayesian Model Averaging
CNN	Convolution Neural Network
EM	Expectation-Maximization
NB	Naive Bayes
NN	Nearest Neighbor
OCSVM	One-class Support Vector Machine
PU	Positive and Unlabeled
SVM	Support Vector Machine

List of Abbreviations

List of Tables

2.1	PU Learning	12
3.1	Positive class proportion validation (Kyoto dataset)	32
3.2	Positive class proportion validations (UMN dataset)	32
3.3	Prior distribution validation (MNIST dataset)	33
3.4	Kyoto dataset: Feedback validation (PN represents performance with fully labeled training data). Note. Reprinted from [2]. Copy- right (2017) by Zhang, Jiaqi.	34
4.1	Experimental datasets	42
4.2	F score of all datasets	44
4.3	Positive class proportion validation (Kyoto dataset)	44
4.4	Positive class proportion validation (UMN dataset)	45
4.5	Prior distribution validation (MNIST dataset)	45

List of Tables

List of Figures

1.1	surveillance monitoring video data.	3
1.2	The structure of this thesis.	6
2.1	Illustration of anomaly detection problem. G1 and G2 are clusters of normal data. F1, F2 and F3 are anomalies.	8
2.2	The data structure of Positive and Unlabeled Learning. Reliable negative data are identified for some previous works.	12
2.3	Illustration of cluster-based Positive and Unlabeled Learning [3, 4]. Note. Reprinted from [4]. Copyright (2011) by Xiao, Yanshan. . . .	14
2.4	Illustration of similarity-based Positive and Unlabeled Learning [5]. Note. Reprinted from [5]. Copyright (2011) by Nguyen, Minh Nhut. . . .	15
3.1	Framework for proposed PU learning method. We first apply a linear model [6] 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 on the remaining unlabeled set U' to predict reliable negative and positive instances and add them to the training pool. Finally, we train the classifier with the reliable data from step 2 until the predicted labels of the remaining unlabeled samples are consistent with the previous iteration.	21
3.2	Illustration of data structure and validation perspective.	30

LIST OF FIGURES

3.3	Performance for different ratios of distribution overlapping for two classes.	34
3.4	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.	35
4.1	Structure for the proposed method.	38
4.2	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.	40
4.3	Performance for different ratios of distribution overlapping for two classes.	46
5.1	Network illustration of heterogeneous categorical events.	48

Chapter 1

Introduction

1.1 Background

As data from various fields are getting easier to be collected, anomaly detection has been playing a more important role in wide domains. It has many applications in bio-medical and security fields. Different types of data structures are involved such as sequence data, text data, image data and graph data. It has wide applications in bio-medical and security fields. Some typical examples of network security are bank fraud, intrusion detection and surveillance monitoring system. As for bio-medical field, applications like system health monitoring, event detection for sensor, and ecosystem disturbances detection also have rapid developments due to anomaly detection.

Anomaly detection is the problem to identify items whose pattern do not conform to an expected pattern or behaves differently with other data in the dataset. Anomaly detection is also referred to as outliers detection, noise removal and novelty detection. However, they are related but distinct from anomaly detection. Anomalies are induced in the data for a variety of reasons, such as malicious activity, they are interesting to the analyst. Noise is not of interest data to the analyst

and hindrance to data analysis while novelty refers to previously unobserved patterns in the data and those patterns will be incorporated into the normal model after being detected.

There are mainly three categorical of techniques for anomaly detection problem. Unsupervised methods detect anomalies based on the assumption that anomalies are the items fit least to the majority of the data. Labels for both normal and anomalies are required for supervised method to train a classifier. Existing semi-supervised methods often work as one-class classification methods which build a model to represent the normal group and the likelihood of testing data generated by the learnt model is treated as the anomaly score.

1.2 Motivation

Anomaly detection is a special case of classification problem where data is dominated by the positive class. Compared to normal classification problem, anomaly detection has the following characteristics and challenges:

- Availability of labeled data for training is a major issue. Anomalies could be different from each other in a number of ways; hence it is difficult to properly label them. Take surveillance monitoring video data as example, anomalies may be defined as unusual crowd activity or loitering individuals or motion in the restricted area. (Fig. 1.1)
- The proportion of anomalies is usually much smaller than normal data, which makes it an unbalanced problem. Different techniques can be adopted for unbalanced problem during pre-processing and training process.
- The inter-class similarity and intra-class variation make it difficult to separate the normal and anomalous behaviors. Anomalies often pretend to act as a normal activity which makes the identification more difficult. Various

Figure 1.1: surveillance monitoring video data.

behaviors and different kinds of anomalies are presented in the data which will increase the difficulty of identification.

In the work of this thesis, we will focus on the challenges mentioned above by proposing a semi-supervised method for anomaly detection.

1.3 Objectives

A hybrid solution is required to tackle the above challenges. There are two major purposes in this thesis.

Firstly, we aim to propose an improved framework based on PU learning method with the linear model [6, 7] as the initial step to tackle the task of anomaly detection. The effective PU learning framework proposed is to tackle the challenge of constructing an anomaly detector using multi-features for unbalanced data when only limited labeled positive data and unlabeled data are provided.

Secondly, an updated boosting PU learning framework need to be designed to tackle the instability of self-learning process. 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.

To summarize, our objectives of this work are three folds:

- 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.
- build a new PU learning framework for an uncommon unbalanced scenario where unlabeled set is dominated by positive data.
- propose to use the characteristic of boosting in PU learning to tackle the instability of self-learning process.

1.4 Contributions

Inspired by the limitations of anomaly detection, we proposed to apply Positive and Unlabeled learning which is a semi-supervised method so that no labeling of anomalies is required in the training process. A self-learning process is adopted in the framework to better separate the normal and anomalous behaviors. To address the unbalanced problem, we use the estimated class prior to decide the different learning speeds of two classes in the self-learning process.

The first proposed framework uses a linear model as a pre-process step to obtain the reliable negative instances. Then self-learning process is utilized to improve the model by iteratively adding reliable instances with different speeds based on the estimated class prior until most of the unlabeled data are classified. Here, the class prior is the proportion of positive samples in the given unlabeled data. The relation between the proportion of remaining unlabeled data and estimated class prior error is also studied to cancel the error of estimated class prior. Final detector is built after labels for the remaining unlabeled data do not change from the previous iteration.

In order to tackle the instability characteristic caused by self learning process

during the training process, we employ boosting in the self-learning. The error rate for each classifier during the self-learning process is estimated to measure the weight for each classifier. Different weights are attached to the classifiers to improve the performance of the combined detector. The boosting structure can be treated as a memory device to decide how much we should memorize at each step.

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 under different settings.

1.5 Organization of the Report

The organization of the thesis is described as follows, which is also shown in Figure 1.2.

In this chapter, we give an introduction of the anomaly detection problem with the motivation inspired by its challenges.

Chapter 2 introduces the literature Review of anomaly detection and Positive and Unlabeled learning.

Chapter 3 introduces a new PU learning framework, so that conventional PU learning methods can be applied to the anomaly detection problem [2]. We first briefly introduce the motivation in Chapter 3.1. Theoretical illustration details are given in Chapter 3.2. We use a novel linear model [6] as pre-training process to extract most reliable anomalies. Then a self-learning process is employed to gradually update the classifier. The positive class prior is estimated to adjust different learning speeds for positive and negative classes. Experiment settings and results are presented in Chapter 3.3.

Figure 1.2: The structure of this thesis.

In Chapter 4, we propose a more powerful framework by integrating PU learning process with boosting procedure to tackle the instability of self-learning. When feedback is available, 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. The theoretical proof of the selection of weights is given in Chapter 4.2. Experimental results are presented in Chapter 4.3 to demonstrate its effectiveness.

Finally, Chapter 5 concludes the report and introduces the future works.

Chapter 2

Literature Review

2.1 Anomaly Detection

Anomaly detection [8] is a technique used to identify unusual patterns whose behavior is not conform to the expected behavior (Fig. 2.1).

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 methods [9–12], clustering based methods [13], nearest neighbor (NN) based methods [14], statistical [15], information theoretic [16] and spectral methods [17]. Zhou *et al.* [9] 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. Li *et al.* [10] introduce a novel transferred deep CNN model that is trained by the difference between pixel pairs generated from the reference image scene. The difference between pixel pairs in the testing images which are constructed by combining the center pixel and its surrounding pixels, is tested by the trained CNN to get the similarity results. Xiao *et al.* [11] propose to use Bayesian Model Averaging (BMA) by averaging over the k-best

Figure 2.1: Illustration of anomaly detection problem. G_1 and G_2 are clusters of normal data. F_1 , F_2 and F_3 are anomalies.

BN classifiers. Mascaro *et al.* [12] combine dynamic and static Bayesian network models to improve the coverage of the overall model. Yang *et al.* [13] use a hyperspherical clustering for extracted spatio-temporal features. In the pre-processing, ViBe is used for background modeling and object detection. Then to track trajectories of people in the video, Kalman filter and Hungarian cost algorithms are adopted. Ghosh *et al.* [14] 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.* [15] directly model the likelihood of heterogeneous categorical events. It is achieved by entity embedding followed by weighted combination of pairwise interactions. Noise-contrastive estimation is utilized to reduce the computational complexity caused by exponentially large event space so that the model can be learned efficiently. Ando [16] analyses anomaly detection from an information theoretic perspective by the averaged atypicality and the complexity of the clusters.

Anomaly detection is a very important topic in security, economics and medi-

cal fields. Besides machine learning methods for common data structure, there are many other methods proposed for anomaly detection of a particular data structure. Many works consider anomaly problem for time-series datasets [18–20]. Jones *et al.* [18] propose to transfer exemplar-based model for spotting anomalies in single dimension of the time series to multiple dimensions by a relation function. Günnemann *et al.* [19] aim for anomalies in time-series of rating distribution by an autoregressive process from latent multivariate base behavior. Koutra *et al.* [20] propose method TENSORSPLAT which use "PARAFAC" decomposition method and tensor analysis to give good insights about the large networks and spot anomalies in multi-aspect data. Nandi *et al.* [21] mine template sequences from execution logs to form a control flow graph (CFG) spanning distributed components which represent the baseline healthy system state. The anomalies are detected by deviations from the expected behavior of runtime logs. Some other works also involve anomaly detection for heterogeneous data [22, 23]. Melnyk *et al.* [22] consider anomaly detection in aviation safety domain of heterogeneous, multivariate, variable-length time series datasets. The framework detects anomalies based on dissimilarities between the prediction produced by the model and data represented by a semi-Markov switching vector autoregressive model. Manzoor *et al.* [23] propose clustering based method StreamSpot for heterogeneity and streaming data by a new similarity function that represents the heterogeneous graph by a vector so that the relative frequency can be compared.

Due to the real situation of many applications, labeled anomaly samples are very difficult to obtain. For instance, labeled insider behaviors are hardly available for training. Anomaly samples are often very rare comparing with normal data and varied from each other. Thus, many works solve anomaly detection based on unsupervised methods [24] and supervised one-class classification methods [25–28]. Amer *et al.* [24] present two modified one-class SVMs: Robust one-class SVMs and eta one-class SVMs for unsupervised anomaly detection. It is achieved

by less contribution from outliers to the decision boundary comparing to normal instances. Li *et al.* [26] propose One-class Very Fast Decision Tree (OcVFDT) algorithm based on VFDT and POSC4.5 so that only some positive samples and unlabeled samples are required for the learning. Song *et al.* [28] propose One-Class Conditional Random Fields (OCCRF) to maximize the margin between normal and anomaly data. They introduce hinge loss into a regularized risk minimization framework.

2.2 Positive and Unlabeled Learning

Positive and Unlabeled (PU) Learning is a kind of semi-supervised method [29–31]. PU learning refers to the problem where only labeled positive data and unlabeled data are provided in the training process (as shown in Fig.2.4). It has been researched within diverse research areas and applications.

For positive and unlabeled learning problem, the true labels Y for training data X is partially provided. The labeled positive set is denoted as P and unlabeled set is denoted as U . The class prior π is defined as the probability of positive instance happening $\pi = p(Y = 1)$. Thus, distribution of unlabeled set can be represented by $q(X, \pi) = \pi p(X | Y = 1) + (1 - \pi)p(X | Y = -1)$.

The feasibility of PU learning can be proved in terms of probability of error [1]. The total probability of error is

$$p(f(X) \neq Y) = p(f(X) = 1 \cap Y = 0) + p(f(X) = 0 \cap Y = 1). \quad (2.1)$$

The probability of error for negative class can be represented as

$$\begin{aligned} p(f(X) = 1 \cap Y = 0) &= p(f(X) = 1) - p(f(X) = 1 \cap Y = 1) \\ &= p(f(X) = 1) - (p(Y = 1) - p(f(X) = 0 \cap Y = 1)). \end{aligned} \quad (2.2)$$

Substituting Eq 2.1 into Eq 2.2, the probability of error has a new representation as

$$p(f(X) \neq Y) = p(f(X) = 1) - p(Y = 1) + 2p(f(X) = 0 \cap Y = 1)p(Y = 1). \quad (2.3)$$

Note that $p(Y = 1)$ is constant. If we are able to hold $p(f(X) = 0 \mid Y = 1)$ small, then learning under PU learning setting is approximately the same as minimizing $p(f(X) = 1)$. If the number of positive examples P and the number of unlabeled examples U are large enough, holding $p(f(X) = 0 \mid Y = 1)$ small while minimizing $p(f(X) = 1)$ can be converted to minimize $p_U(f(X) = 1)$ while hold $p_P(f(X) = 1) \geq recall$.

Feasibility of PU learning can also be proved in terms of distribution of data. Assume P_0 and P_1 is the conditional distribution of negative and positive classes. We denote the conditional distribution of unlabeled data as P_U , thus P_U can be represented as

$$U \sim P_U = (1 - \pi)P_0 + \pi P_1. \quad (2.4)$$

If P_1 is divide on both sides, we can obtain

$$\frac{P_U}{P_1} = (1 - \pi)\frac{P_0}{P_1} + \pi. \quad (2.5)$$

$\frac{P_U}{P_1}$ is changing as a linear relationship with $\frac{P_0}{P_1}$. Hence, estimator for the conditional probability of positive and unlabeled data should also be applicable to discriminate positive data from negative data in theory.

Due to the characteristic of PU learning, it is suitable for many real world

Figure 2.2: The data structure of Positive and Unlabeled Learning. Reliable negative data are identified for some previous works.

Table 2.1: PU Learning

Applications	Based on spy technique [1]	Similarity based
Text classification	[1, 3, 4, 32, 33, 41–44]	
Disease gene classification		[35]
Image classification		[38]
Data stream problem	[40]	[5]
Applications	Linear classifier based	Possibility based
Text classification		[45]
Disease gene classification		
Image classification	[6, 7]	[37]
Data stream problem		[46]

applications in text detection [1, 3, 32, 33], bioscience [34–36] and image classification [6, 37–39]. Nowadays it is also applied to data stream problem [5, 40].

2.2.1 Methods based on spy technique [1]

According to the name of Positive and Unlabeled Learning, the provided labeled positive data are the object or measure target. The unlabeled data are a large group that rarely contains the object data (positive data). Thus, many papers start learning from regarding all the unlabeled data as negative set to get the initial classifier. Work by Liu in [1] and [32] are very important traditional PU learning

methods that many methods are proposed based on them. Liu *et al.* [1] first regard the unlabeled set as negative class and adopt the spy technique. Some positive instances are introduced to the unlabeled set to get the threshold score between two classes. Then the classifier is built based on Expectation-Maximization (EM) using provided positive instances and the extracted negative instances. Li and Liu [32] propose an effective classifier which start by regarding unlabeled samples as negative. The Rocchio method with clustering is combined with SVM technique to identify the reliable negative instances.

Most of other papers propose their methods based on spy technique [1,32]. The positive class which is the assigned target for these methods has a small quantity in unlabeled data, which is different from the situation of the anomaly detection problem. They can start the training by regarding the unlabeled set as negative set. Ren *et al.* [3] first use similar way of [1] to identify some negative instances from unlabeled set. Representative positive examples and negative examples are generated based on Latent Dirichlet Allocation. To build the final classifier, two similarity weights are assigned to the remaining unlabeled instances and incorporated into modified SVM. Li *et al.* [33] use spy algorithm followed by EM algorithm with Naive Bayes (NB) classifier to get final labels, which is very similar to [1]. Xiao *et al.* [4] 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 instances with similarity weight concerning local information and similarity weight concerning global information, to build an SVM-based final classifier. Nigam *et al.* [42] 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.* [43] first use spy technique to select the samples with higher in-target-domain probability. The calibrated probabilities are utilized as sampling weights to train an instance-weighted

Figure 2.3: Illustration of cluster-based Positive and Unlabeled Learning [3, 4]. Note. Reprinted from [4]. Copyright (2011) by Xiao, Yanshan.

Naive Bayes model. Fusilier *et al.* [44] 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. Mordelet *et al.* [41] iteratively train many binary classifiers that can discriminate the known positive examples from random subsamples of the unlabeled set, and average the 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 which indicates low contamination by positive instances, then a better classifier will probably be obtained than if it has high contamination. Hence, it is not appropriate in the case of anomaly detection, where positive samples dominate the unlabeled set.

2.2.2 Similarity or clustering based methods

Similarity based or cluster based methods are popular for PU learning. Li *et al.* [5] first use PCA and k-means to do local clustering and feature selection. Then

Figure 2.4: Illustration of similarity-based Positive and Unlabeled Learning [5]. Note. Reprinted from [5]. Copyright (2011) by Nguyen, Minh Nhut.

those clusters which have large distance from positive class are regarded as reliable negative clusters, and finally the other ambiguous clusters are added gradually. Yang *et al.* [35] partition unlabeled data into four sets based on their probabilities of belonging to positive class. Weighted SVM is used to build a multi-level classifier based on provided positive set and four unlabeled sets. Yang *et al.* [38] use the cosine similarity and threshold criteria to get the potential negative samples and adopt Rocchio algorithm to extract the final reliable negative set. Then classifiers are iteratively applied to get the final classifier.

2.2.3 Linear classifier based methods

Some papers work on theoretical conclusions for PU learning. Niu *et al.* [47] theoretically compare PU and NU learning against the training when all labels are provided (PN learning). They prove that given infinite unlabeled data, PU or NU learning will almost always performs better than PN learning. Du Plessis *et al.* [7] prove that PU learning problem can be solved by cost sensitive learning and non-convex cost function should be used to avoid the intrinsic bias of convex

surrogate loss function. Based on previous theoretical proof and further refinement, du Plessis *et al.* [6, 7] propose novel idea to solve PU learning by only a linear classifier so that no iteration is needed. Different convex cost functions are used for positive set and unlabeled set. There is no class prior constrain in the training process but they assume class prior is provided. For our proposed PU learning methods, we assume class prior is not provided and the training is iterative process which will have better performance comparing with [6, 7]. Du Plessis *et al.* [48, 49] also propose methods to estimate the class prior under PU learning settings. The ideas are achieved by penalized divergences to cancel the effect of absence of negative examples. [49] further demonstrates the computationally efficient of using penalized L_1 -distance and provides the analysis of its uniform deviation bound and estimation error bound.

2.2.4 Probability based methods

Li *et al.* [37] and Elkan *et al.* [45] developed a classifier based on positive and unlabeled examples to predict the probabilities of being positive. They show that the predicted probabilities and true conditional probabilities can be related by only a constant factor.

2.2.5 Streaming based methods

For data streaming situation, the papers propose their model based on the specific situation. Li *et al.* [46] is the first to apply PU learning to data streaming problem. It focuses on improving the speed and algorithm efficiency. It uses classifiers like Rocchio classifiers and k-means based on the spy extraction to implement in linear time, so that the main time complexity is from the final SVM classifier. Chang *et al.* [40] propose a principled probabilistic approach for real-world systems like Twitter following and Facebook like. It models the situation for real life network connection

and gives the update formula to predict the future connections.

Chapter 3

Positive and Unlabeled Learning for Anomaly Detection

3.1 Introduction

As it is difficult to get labels for anomalies in anomaly detection problem, only labeled normal data and some unlabeled data are available for training in many applications. The challenge can be tackled by PU learning, where only labeled positive samples and unlabeled samples are required in the training process. Anomalies are regarded as negative class to avoid labeling of anomalies in PU learning. 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 fully use of the provided data. In addition, PU learning can avoid labeling of anomalies comparing to supervised two-class classification methods.

For conventional PU learning methods, positive data are the detection target and the unlabeled instances are mostly negative data [1, 32]. The traditional methods usually regard unlabeled data as negative to build the initial classifier. However, for anomaly detection problem, the detection target anomalies are regarded as negative to avoid labeling. Labeled positive instances are normal data and anomalies are still the minority in unlabeled set. The positive data are con-

taminated, which will influence the performance. Thus, conventional PU learning methods cannot be applied to anomaly detection problem. In the literature, the proportion of positive instances in unlabeled set is termed as *positive class prior*. To have a more general model, du Plessis *et al.* [6, 7] propose to use a linear model for positive and unlabeled problem. Although with theoretical guarantees of the performance, the linear model is still not good enough since inter-class similarities are common [50], *i.e.*, in many cases the normal data and anomalies are not clearly differentiable. To improve the robustness of our method, we introduce a self-learning process, which works by iteratively extracting reliable positive and negative instances from unlabeled set.

We elaborate the training process of our proposed framework in the following. We first use the linear model [6] 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 the self-learning process. Finally we utilize the updated sets P' , N' and the remaining U' from previous step, classify the remaining unlabeled data to the two classes and train the model until the predicted labels of the remaining unlabeled samples in the $(t + 1)$ th iteration are consistent with that of the (t) th iteration. The pipeline of the proposed method is shown in Fig. 3.1.

The work of Chapter 3 was previously published in [2].

3.2. PROPOSED APPROACH: PU LEARNING FOR ANOMALY DETECTION

Figure 3.1: Framework for proposed PU learning method. We first apply a linear model [6] 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 on the remaining unlabeled set U' to predict reliable negative and positive instances and add them to the training pool. Finally, we train the classifier with the reliable data from step 2 until the predicted labels of the remaining unlabeled samples are consistent with the previous iteration.

3.2 Proposed Approach: PU Learning for Anomaly Detection

Let $X = \{x_1, \dots, x_i, \dots, x_n\} \in R^{m \times n}$ denote the training data. There are m pieces of training data with n dimensions. $Y = \{y_1, \dots, y_i, \dots, y_n\} \in \{-1, 1\}$ 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 $\pi = p(Y = 1)$ 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

$$q(X, \pi) = \pi p(X \mid Y = 1) + (1 - \pi) p(X \mid Y = -1) \quad (3.1)$$

3.2.1 Pre-training using Linear Model

We are only provided with labeled positive set and unlabeled set, thus, the first step is to extract the most reliable negative set N from the unlabeled set U . To

address the problem of conventional PU learning for anomaly detection, we use a linear model $g = a^T \psi(x) + b$ and apply different cost functions for positive set P and unlabeled set U [6]. Here $\psi(X) = [\psi_1(X) \dots \psi_m(X)]$ is a set of basis 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

$$J(g) = \pi E_1[l(g(U))] + (1 - \pi) E_{-1}[l(-g(U))] \quad (3.2)$$

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

$$\min_{g \in G} J(g). \quad (3.3)$$

Due to the data distribution of training data as Eq 3.1, we have

$$E_U[l(-g(U))] = \pi E_1[l(-g(U))] + (1 - \pi) E_{-1}[l(-g(U))]. \quad (3.4)$$

Substitute Eq 3.4 into Eq 3.2, the expected misclassification rate of training data can be represented by positive data and unlabeled data [6]

$$J(g) = \pi E_1[l(g(U)) - l(-g(U))] + E_U[l(-g(U))]. \quad (3.5)$$

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 $l(z)$ as squared loss, the composite loss can be normalized as $-z$, so that misclassification rate of training data can be further simplified as [6]

$$J_s(g) = \pi E_1[-g(U)] + E_U[l_s(-g(U))]. \quad (3.6)$$

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

3.2. PROPOSED APPROACH: PU LEARNING FOR ANOMALY DETECTION

---

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

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

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

$$\pi = \arg \min_{0 \leq \theta \leq 1} \int f\left(\frac{q(x; \theta)}{p(x)}\right) p(x) dx \quad (3.7)$$

with penalized L_1 distance [48]. Partial model

$$q(x, \pi) = \pi p(x | y = 1) \quad (3.8)$$

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

The parameter α of linear model indicates how discriminative of each feature. Those with high value of α are more important in the detection task. The characteristic of linear model leads to good performance under multi-features settings. At the end of the first step by the 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.

3.2.2 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, $p\%$ of the unlabeled set are divided based on the estimated positive class prior. The

remaining $(1 - p\%)$ of the unlabeled set U resulted from *Algorithm1* are kept for the last step to correct the error in positive class prior estimation. The selection of ratio $p\%$ is derived in the next section (Chapter 3.2.3).

Algorithm 1 Self-learning process

Input: Training set: P', U', N' , positive class prior π , $speed_n = K$, $\%p$

Output: Updated P', U' and N'

```

1:  $\delta^n = K$ ,  $\delta^p = K \times \pi / (1 - \pi)$ 
2: model  $M_2 = Classifier(P, N)$ 
3:  $score = predict(M_2, U')$ 
4:  $N' = N$ 
5: for  $|N'| < p\% \times (1 - \pi) \times |U|$  do
6:    $I = rank(score)$  in ascending order
7:    $Nadd = \{x_i | (score(i) < I(K))\}$ 
8:    $Padd = \{x_i | (score(i) > I(end - K \times \pi / (1 - \pi) + 1))\}$ 
9:    $N' = N' \cup Nadd$ ,  $P' = P' \cup Padd$ ,  $U' = U' - Nadd - Padd$ 
10:  model  $M_2 = Classifier(P', N')$ 
11:   $score = predict(M_2, U')$ 
12: end for

```

3.2.3 Final Detector Building

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. We propose *Algorithm2* to determine more accurate labels for the remaining part of the unlabeled data.

Initially, instances in set U' are unlabeled. At the end of the first iteration, each instance will be assigned a temporal label and will be classified as positive or

negative class. In subsequent iterations, a new model will be generated based on P_{temp} and N_{temp} . Then the new model will be applied to U' and each instance in set U' will be assigned with a new label. The process continues until the labels for instances of U' in the $(t + 1)$ th iteration are consistent with that of the (t) th iteration and the final detector is built. Here it should be noticed that instances P_{pred} and N_{pred} extracted from U are temporarily used to form new set P_{temp} and set N_{temp} , instead of adding to set P' and set N' . They will be returned to U' after obtaining the new labels. The instances in U' will only be added to set P' and set N' when their labels are consistent with the labels from the previous iteration. After this final training step, the whole unlabeled data will be divided into set P' and set N' and a final anomaly detector is built. As such, all given information of positive set and unlabeled set are fully utilized to form the final model for anomaly detection or other applications.

The relationship between the error of estimated positive class prior and the proper $p\%$ 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 $(1 - p\%) \times size(U)$ 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:

$$p\% < \begin{cases} \frac{1-\pi}{1-\pi(e+1)} & \text{when } e < 0, \\ \frac{1}{1+e} & \text{when } e > 0. \end{cases} \quad (3.9)$$

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 $p\%$ is picked to avoid both situations. Thus the relationship between the error of

estimated positive class prior and positive class prior $p\%$ that needs to be satisfied is:

$$p\% < \begin{cases} \frac{1-\pi}{1-\pi(e+1)} & \pi > 0.5, \\ \frac{1}{1+e} & \pi \leq 0.5. \end{cases} \quad (3.10)$$

In the anomaly detection scenario, the positive class prior is much larger than 0.5.

Algorithm 2 build final anomaly detection method

Input: Training set: P', U', N'

Output: anomaly detection model

```

1:  $t = 1$ 
2:  $label_{U'}(t) = predict(M_2, U')$ 
3:  $label_{U'}(t-1) = \sim label'_U(t)$ 
4: while  $label_{U'}(t) \neq label_{U'}(t-1)$  do
5:    $N_{pred}$  =instanced with label 0,
6:    $P_{pred}$  =instanced with label 1.
7:    $N_{temp} = N' \cup N_{pred}, P_{temp} = P' \cup P_{pred}$ 
8:   model  $M_3 = Classifier(P_{temp}, N_{temp})$ 
9:    $t = t + 1$ 
10:   $label_{U'}(t) = predict(M_3, U')$ 
11: end while

```

3.3 Experiment

3.3.1 Evaluation Measures

We use information retrieval measures F score for our target of anomaly detection. F score considers both the precision and the recall of the test results. Since anomaly is our target, whose performance should be measured and compared, the ‘positive P' in the measurement is regarded as anomaly. It is different from training process where normal data are regarded as positive. Thus, our method is actually ‘Negative and Unlabeled Learning’ for anomaly detection scenario. We use P^* to denote the anomalies in the following formulas. If the classifier is represented by F , $F(X)$ denotes the predicted labels. The measurement formulas are listed as follows:

$$\begin{aligned}
 Precision &= \frac{p(F(x) = 1 \cap y = 1)}{p(y = 1)} = \frac{TP^*}{(TP^* + FP^*)} = \frac{TP^*}{\text{all detected anomalies}} \\
 Recall &= \frac{p(F(x) = 1 \cap y = 1)}{p(F(x) = 1)} = \frac{TP^*}{(TP^* + FN^*)} = \frac{TP^*}{\text{all real anomalies}} \\
 Fscore &= \frac{2 \times Precision \times Recall}{(Precision + Recall)}
 \end{aligned}$$

3.3.2 Experiment Dataset and Settings

The evaluated datasets are from diverse fields of anomaly detection: Cyber security (Kyoto dataset); Image dataset (MNIST dataset); Video dataset (UMN dataset); and also a synthetic dataset. Random Forest has the best performance and is selected among Random Forest, Naive Bayes and decision tree since it is an ensemble learning method with both tree bagging and feature bagging process. The feature bagging can achieve to extract strong predictors by selecting the features in many of the trees with a random subset of features, which will lead to good performance under multi-features problem. The basic function $\psi(X)$ is chosen as Gaussian function and $p\%$ is selected as 90% for all four datasets. The speed K is selected based on the performance of the initial classifier and the number of instances in the unlabeled set. We conduct grid-search experiments to find the parameter K for different datasets. K is selected as 1000 for Kyoto dataset, 20 for UMN dataset, MNIST dataset and the synthetic dataset.

- Synthetic dataset: The synthetic dataset is generated based on Gaussian distribution to vary the distribution overlaps of normal data and anomalies. We simulate 2-dimensional data with distribution characters of anomaly detection datasets. Negative data or anomalies are multi-clusters with larger variation while variation is smaller for normal data. The distributions of positive and negative classes have overlaps and thus the ratio of distribution overlaps will affect the performance of anomaly detection. 600 instances are

randomly selected for each training set.

- **Kyoto dataset:** Kyoto dataset is generated by the simulation over a virtual network for intrusion detection field. Intrusion detection is also a real application of anomaly detection whose common fact is that it is very difficult to label an intrusion while the normal users are easy to obtain. 21 out of 24 statistical features are utilized. The intrusions are regarded as negative class and normal users are regarded as positive class in the training process. Due to the huge amount of data, we only randomly select 100,000 samples each time and show the average performance.
- **UMN dataset:** UMN dataset is a video dataset for abnormal crowd activities detection. The video is split into 16 frames long clips with a 15-frame overlap between two consecutive clips and each clip is labeled as normal or abnormal. The spatial-temporal features of 64×64 dimensions for each video clip are learnt with 3D convolutional network [51]. 3000 video clips are randomly chosen for training and average results of 10 groups are shown.
- **MNIST dataset:** We use the same experimental setup as [6] for MNIST dataset. The images for ‘0’ digit are treated as positive class, and images for other digits are treated as negative class. Under same class prior setup, the precision, recall and F score for all 9 negative classes (i.e., one negative classes for each digit ‘1’,...,‘9’) will be averaged. Dimensionality is reduced to two via principal component analysis. 200 positive samples and 400 unlabeled samples are used for training each time.

As shown in Fig. 3.2,

$$\begin{aligned} \% \text{ of } P &= \frac{P}{P \cup U} \\ \text{class prior } \pi &= \frac{\text{Positive data in } U}{U} \end{aligned}$$

Figure 3.2: Illustration of data structure and validation perspective.

3.3.3 Existing Methods for Comparison

- One-class SVM [52]: Only the labeled positive set P is utilized to build the one-class Support Vector Machine (SVM) model. The data we use are all with correct labels in positive class. However, it cannot utilize the information in unlabeled data.
- S-EM [1]: It is a widely used PU learning method which uses spy-positive instances to identify most likely negative instances from the unlabeled data. Then EM algorithm is used to obtain the final classifier.
- Convex Linear model [6]: It is a linear-in-parameter model for positive and unlabeled problem. It uses different loss functions for positive and unlabeled dataset. The loss function is chosen as squared loss function $l_s(z) = \frac{1}{4}(z-1)^2$ which is the same as our step 1 (Chapter 3.2.1).
- Same classifier with fully labeled training data (represented by PN): With this setting, all labels for original unlabeled class 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.

3.3.4 Experiment Results and Discussion

Positive class proportion validation

The validation is needed to show how the proportion of the provided positive set will affect the training performance. The Kyoto dataset and UMN dataset are used for the positive class proportion validation. The ‘% of P ’ in Table 3.1 and 3.2 refer to the number of instances in provided positive set P divided by all number of all training instances (illustrated by Fig. 3.2). The percentage of positive set in training dataset varies from 20% to 80% for Kyoto dataset and 30% to 60% for UMN dataset.

The F scores of Kyoto dataset and UMN dataset under various ‘% of P ’ are shown in Table 3.1 and 3.2. Methods of One-class SVM, S-EM [1], Convex PU [6] and PN are also compared. S-EM [1] cannot work for the video dataset. Our proposed method performs better than existing methods under different ‘% of P ’ if only positive and unlabeled data are provided. The proposed method has comparable performance with the model built by the same classifier with labels for all training data. As shown in table 3.1 and 3.2, performance of the proposed method will not be affected by variations of positive class proportions. It can perform well even when provided positive class proportion is only 20%.

From the parameter α of linear model and random forest, we could identify the features with more discriminative information. The strong predictors for Kyoto dataset are: session start time, percentage of connection that ‘SYN’ errors in Dst_host_srv_count feature, the connection’s service type and percentage of connection that have ‘SYN’ errors in Dst_host_count feature [53].

Prior distribution validation

For anomaly detection problem in real applications, the number of anomalies is much smaller than normal data. The experiment is conducted to validate that our

Table 3.1: Positive class proportion validation (Kyoto dataset)

% of P	OCSVM	S-EM [1]	LPU [32, 54]	Convex Linear model [6]	Ours1	PN
20%	80.13%	58.21%	84.43%	89.46%	92.17%	98.81%
40%	75.70%	59.26%	81.85%	87.69%	91.79%	98.65%
60%	80.21%	58.61%	81.22%	88.13%	92.29%	98.30%
80%	77.62%	73.49%	85.47%	85.95%	89.14%	95.99%

Table 3.2: Positive class proportion validations (UMN dataset)

% of P	OCSVM	S-EM [1]	LPU [32, 54]	Convex Linear model [6]	Ours1	PN
30%	28.62%	26.01%	26.01%	92.55%	93.43%	97.75%
40%	28.96%	25.43%	25.43%	86.69%	92.24%	97.07%
50%	30.15%	25.81%	25.81%	86.96%	95.02%	96.77%
60%	30.72%	25.49%	25.49%	92.30%	94.50%	96.35%

method performs better under different class priors especially in such unbalanced situation. The ‘class prior π ’ in Table 3.3 is the ratio of positive data in given unlabeled set U (illustrated by Fig. 3.2). MNIST dataset is used for the class prior distribution validation. The class priors are varied from 0.1 to 0.9 across the experiments and ‘% of P ’ is set as 50%. The performance of MNIST dataset under different class priors is shown in Table 3.3. Methods of One-class SVM, S-EM [1], Convex PU [6] and PN are also compared. F score of our proposed method outperforms existing benchmark methods under different class prior situations if only positive and unlabeled data are given. The proposed method has comparable performance with the situation when all labels for training data are given. The performance improvement comparing with other methods is obvious in unbalanced situations. The results of one-class SVM will be omitted because its F scores are around 65% which are low comparing to other methods. From Table 3.3, we can tell that the increasement of recall causes the increasement of F score. That means our proposed framework decreases the number of false anomaly detection, which

3.3. EXPERIMENT

Table 3.3: Prior distribution validation (MNIST dataset)

positive class prior π	OCSVM	S-EM [1]	Convex Linear model [6]	Ours1	PN
0.1	83.92%	68.27%	97.10%	94.74%	95.82%
0.2	84.69%	67.67%	94.52%	94.32%	95.20%
0.3	82.21%	67.16%	90.00%	93.34%	92.99%
0.4	83.92%	66.57%	84.96%	92.99%	92.38%
0.5	79.67%	67.65%	77.06%	91.53%	90.63%
0.6	82.04%	68.08%	78.92%	91.27%	89.53%
0.7	78.04%	69.50%	75.39%	87.11%	85.50%
0.8	76.68%	68.79%	74.59%	87.77%	85.41%
0.9	66.05%	65.00%	64.60%	79.22%	85.41%

is the number of normal data regarding as anomalies.

Distribution overlapping ratio validation

This validation test is needed to prove that the proposed method can perform better compared to existing methods when normal data and anomalies have different overlapping ratios. To achieve the experimental settings with different overlapping ratios of two distributions, we synthesize a dataset based on a 2-dimensional Gaussian distribution, which makes it easier to model the distributions overlaps. Distribution overlapping ratios of normal data and anomalies are varied from 10% to 45%. The class prior is set at 90%, which means 90% of the training data are normal data. 200 positive data points and 400 unlabeled data points are sampled as training data. Methods of One-class SVM, S-EM [1], Convex PU [6] and PN are compared and shown in Fig. 3.3. As expected, the F score of all methods will decrease with the increasing of distribution overlapping ratio. Performance of Convex PU [6] decreases faster when overlapping ratio increases compared with other methods. The proposed method has the best performance when overlapping ratio is less than 45% and methods like one-class SVM will catch up when overlapping ratio exceeds 45% which is not representative scenario. Detection results for different methods are shown in Fig. 3.4 explicitly.

Figure 3.3: Performance for different ratios of distribution overlapping for two classes.

Table 3.4: Kyoto dataset: Feedback validation (PN represents performance with fully labeled training data). Note. Reprinted from [2]. Copyright (2017) by Zhang, Jiaqi.

% of P	Ours1	Ours1 with feedback	PN
20%	94.79%	94.86%	99.04%
40%	94.40%	94.62%	98.41%
60%	94.53%	94.46%	98.71%
80%	92.73%	93.83%	94.91%

Feedback validation

In many real world applications such as intrusion detection problem, we can have feedback information for the extracted intrusions during the training process if there is enough budget. In principle, more feedback provided will lead to a better model. If there is enough budget for testing of all detection results, all the labels are provided at the end of training procedure.

The Kyoto dataset is used for the feedback validation. We compare the proposed method where no feedback and feedback for 50% data is provided. PN method is also compared under different percentages of given positive data in Table 3.4. The performance proves the theory that higher feedback will have better performance; however, the F score of 50% feedback is closer to the proposed method without feedback. Hence, the proposed method can obtain satisfactory performance.

3.3. EXPERIMENT

Figure 3.4: 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.

Chapter 4

Boosting Positive and Unlabeled Learning for Anomaly Detection

4.1 Introduction

In the previous chapter, 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 Chapter 3 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 increase. The increase of steps will also lead to a 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 of decreasing self-learning step size is not guaranteed.

To address the instability of 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 weight for the classifier in each self-learning step. The boosting structure in

Figure 4.1: Structure for the proposed method.

the self-learning process is like a memory device, and the weight of each classifier is to decide how much we should memorize at each step. The main structure of proposed PU learning with boosting method is illustrated in Fig. 4.1. ¹

4.2 Boosting in self-learning process when feedback is available

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:

$$\begin{aligned}
 C_t(x_i) &= \alpha_1 c_1(x_i) + \dots + \alpha_{t-1} c_{t-1}(x_i) + \alpha_t c_t(x_i) \\
 &= C_{(t-1)}(x_i) + \alpha_t c_t(x_i)
 \end{aligned} \tag{4.1}$$

---

¹The work of this chapter has been submitted to IEEE Transactions on Multimedia under title “Boosting Positive and Unlabeled Learning for Anomaly Detection with Multi-features”.

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

$$E = \sum_{i=1}^N e^{-y_i C_t(x_i)} \quad (4.2)$$

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

$$\omega_i^t = \begin{cases} 1 & t = 1, \\ e^{-y_i C_{t-1}(x_i)} = \omega_i^{t-1} e^{-y_i \alpha_{t-1} c_{t-1}(x_i)} & t > 1. \end{cases} \quad (4.3)$$

Substitute Eq 4.2 into Eq 4.3, we have

$$\begin{aligned} E &= \sum_{i=1}^N \omega_i^{(t)} e^{-y_i \alpha_t c_t(x_i)} \\ &= \sum_{i=1}^N \omega_i^{(t)} e^{-\alpha_t} + \sum_{y_i \neq c_t(x_i)} \omega_i^{(t)} (e^{\alpha_t} - e^{-\alpha_t}) \end{aligned} \quad (4.4)$$

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

$$\frac{dE}{d\alpha_t} = \sum_{y_i \neq c_t(x_i)} \omega_i^{(t)} e^{\alpha_t} - \sum_{y_i = c_t(x_i)} \omega_i^{(t)} e^{-\alpha_t} = 0 \quad (4.5)$$

It gives α_t as

$$\alpha_t = \frac{1}{2} \ln \left(\frac{\sum_{y_i = c_t(x_i)} \omega_i^{(t)}}{\sum_{y_i \neq c_t(x_i)} \omega_i^{(t)}} \right) \quad (4.6)$$

From Eq 4.3, we can update α_t in Eq 4.6 by

$$\omega_i^t = \begin{cases} \omega_i^{t-1} e^{-\alpha_{t-1}} & y_i = c_{t-1}(x_i), \\ \omega_i^{t-1} e^{\alpha_{t-1}} & y_i \neq c_{t-1}(x_i). \end{cases} \quad (4.7)$$

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

Figure 4.2: 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.

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 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 4.7, 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 4.6. Thus, we have the updated boosted classifier for the prediction of the 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 the algorithm 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 de-

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. 4.2. 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.

Algorithm 3 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}

```

1:  $\delta^n = K, \delta^p = K \times \pi / (1 - \pi)$ 
2:  $\omega_1 \leftarrow ones(|U|, 1)$ 
3: for  $t = 1 : (r\% \times (1 - \pi) \times |U| - |N'|) / K + 2$  do
4:    $c_t = Classifier(P', N')$ 
5:   give feedback to random  $\%p$  of  $N_{add}, P_{add}$  to get estimated error rate
6:    $\alpha_t = \frac{1}{2} \ln \left( \frac{\sum_{y_i=c_t(x_i)} \omega_i^{(t)}}{\sum_{y_i \neq c_t(x_i)} \omega_i^{(t)}} \right)$ 
7:   update  $\omega_{t+1}$  as Eq 4.7 for all  $x_i$  based on the estimated error rate
8:    $C_t = \alpha_1 c_1 + \dots + \alpha_{t-1} c_{t-1} + \alpha_t c_t$ 
9:    $score = predict(C_t, U')$ 
10:  if  $t < (r\% \times (1 - \pi) \times |U| - |N'|) / K + 1$  then
11:     $S = rank(score)$  in ascending order
12:     $N_{add} = \{x_i | (score(i) < S(K))\}$ 
13:     $P_{add} = \{x_i | (score(i) > S(end - K \times \pi / (1 - \pi) + 1))\}$ 
14:  else if  $t = (r\% \times (1 - \pi) \times |U| - |N'|) / K + 1$  then
15:     $N_{add} = \{x_i | (score(i) < 0)\}$ 
16:     $P_{add} = \{x_i | (score(i) > 0)\}$ 
17:  else
18:    break
19:  end if
20:   $N' = N' \cup N_{add}, P' = P' \cup P_{add}, U' = U' - N_{add} - P_{add}$ 
21: end for
```

4.3 Experiment

4.3.1 Experiment Dataset and Settings

The same evaluation measurement F score and same training data structure (Fig. 3.2) with Chapter 3 are used. Apart from the previously introduced datasets,

Table 4.1: Experimental datasets

Dataset	#Features	#classes	positive class prior
Kyoto dataset	20	2	22%
MNIST dataset	784	10	10%,20%,...,90%
UMN dataset	64×64	2	90%
Statlog (Shuttle) dataset	9	2	73%
Statlog (Landsat Satellite) dataset	36	2	63%
Synthetic dataset	2	2	90%

Dataset	#training instances	positive set % in training set
Kyoto dataset	100,000	20%,40%,60%,80%
MNIST dataset	6000	50%
UMN dataset	3000	30%,40%,50%,60%
Statlog (Shuttle) dataset	32732	15%
Statlog (Landsat Satellite) dataset	3717	27%
Synthetic dataset	600	30%

performances for Statlog (Shuttle) dataset and Statlog (Landsat Satellite) dataset are also verified.

- **Statlog (Shuttle) dataset:** Statlog (Shuttle) dataset is a multi-class classification dataset with 9 numerical attributes. Class 2, 3, 5, 6, 7 which are the smallest five classes are combined to form the outliers class. Data from class 1 are regarded as normal data. 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 3×3 neighborhoods of the satellite images. There are 7 classes in the dataset. Class 2, 4, 5 which are the smallest three classes are combined to form the outliers class. All the other classes are combined to form the positive set with normal data.

Other details of all the datasets are listed in Table 4.1.

4.3.2 Existing Methods for Comparison

Apart from the methods mentioned in the previous chapter, One-class SVM [52], S-EM [1], Convex Linear model [6] and same classifier with fully labeled training data (denoted by PN), we also compare our method with method LPU [32, 54]: It is a widely used PU learning method for text learning. It first regards all the unlabeled data as negative and uses 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 identified during the process.

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

4.3.3 Experiment Results and discussions

F scores for the testing set of the six datasets are shown in Table 4.2. The experimental settings are shown as Table 4.1 and the special selection for each dataset is indicated in Table 4.2. 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 4.2 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 [41] is also investigated. 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.

Positive class proportion validation

The experiment is conducted with the same settings as in Chapter 3. Average experimental results of 10 sets are shown in Table 4.3 and Table 4.4. We can see

Table 4.2: F score of all datasets

F score	OCSVM	S-EM [1]	LPU [32, 54]
Kyoto dataset (% of $P = 40\%$)	75.70%	59.26%	81.85%
MNIST dataset ($\pi = 0.9$)	66.05%	65.00%	0%
UMN dataset (% of $P = 30\%$)	28.62%	26.01%	26.01%
Statlog (Shuttle) dataset	57.82%	27.76%	0%
Statlog (Landsat Satellite) dataset	58.86%	45.92%	0%
synthetic dataset (distribution overlapping=25%)	28.54%	32.34%	25.96%

Convex Linear model [6]	Ours1	Ours2	PN
87.69%	91.79%	94.86%	98.64%
64.60%	79.22%	80.54%	72.18%
92.55%	93.43%	94.46%	97.75%
65.20%	73.92%	82.91%	99.07%
6.71%	32.90%	62.98%	89.54%
24.63%	60.04%	63.31%	64.23%

Table 4.3: Positive class proportion validation (Kyoto dataset)

% of P	Ours1	Ours2	PN
20%	92.17%	94.34%	99.04%
40%	91.79%	94.86%	98.41%
60%	92.29%	94.64%	98.71%
80%	89.14%	94.52%	94.91%

that the performance will not be influenced significantly when the proportion of positive set changes. The updated method outperforms the previous method with the same training settings under different portions of positive set P in the training set.

Positive class prior distribution validation

Same experimental settings of Chapter 3 are kept for the validation. Average results of 10 sets are shown in Table 4.5. The performance of all listed methods will decrease when positive class prior increases. This is because when positive data

4.3. EXPERIMENT

Table 4.4: Positive class proportion validation (UMN dataset)

% of P	Ours1	Ours2
30%	93.43%	94.46%
40%	92.24%	92.98%
50%	95.02%	94.38%
60%	94.50%	94.46%

Table 4.5: Prior distribution validation (MNIST dataset)

positive class prior π	Ours1	Ours2
0.1	94.74%	95.05%
0.2	94.32%	94.48%
0.3	93.34%	93.38%
0.4	92.99%	93.04%
0.5	91.53%	90.95%
0.6	91.27%	91.40%
0.7	87.11%	87.71%
0.8	87.77%	87.70%
0.9	79.22%	87.70%

portion of 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 4.5, it is clear that the performance of updated PU learning with boosting method is increased comparing to the previously proposed method in Chapter 3. The performance improvement is larger when positive class prior is very large, i.e. in anomaly detection scenario.

Distribution overlapping ratio validation

Same experimental settings of Chapter 3 are kept for the validation and the average performance is shown in Fig. 4.3. Larger overlapping ratios of two classes distributions signify how likely anomalies behave like normal instances. The curves in Fig. 4.3 indicate that the performance of proposed Boosting PU learning method is better than previously proposed PU method and all the other methods under

CHAPTER 4. BOOSTING POSITIVE AND UNLABELED LEARNING FOR ANOMALY DETECTION

Figure 4.3: Performance for different ratios of distribution overlapping for two classes.

the same training settings.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, a new framework based on Positive and Unlabeled Learning for anomaly detection problem in a challenging scenario is discussed, where labeled anomaly instances and class prior are not provided.

In the initial work, we proposed a new PU learning framework for anomaly detection. We first use a linear model to identify reliable anomaly instances. Then we use a classifier to gradually add most reliable instances based on estimated class prior during the self-learning process. We build the final anomaly detector when the labels for the last portion of unlabeled class can be reliably determined. We show theoretically the relation between the accuracy of the estimated class prior and percentage of remaining unlabeled data in the last step. Extensive experiments show that our method achieves the state-of-the-art performance comparing to existing methods under different class priors and different distribution overlapping ratios of two classes.

In the subsequent work, to the instability of self-learning process, a new framework is proposed to incorporate the boosting process in the self-learning process to get appropriate weight for each classifier of the self-learning process. Experiments on two new datasets are conducted under different settings. The new PU learning

Figure 5.1: Network illustration of heterogeneous categorical events.

with boosting framework outperforms the previously proposed PU learning method and other existing methods given the same training data settings.

To summarize, the research work addressed the main challenges of anomaly detection by applying PU learning. Two effective frameworks are developed. Promising results on datasets from different areas are shown to verify the performance of proposed methods.

5.2 Future Work

Some possible future research aspects are listed as follows: PU learning for anomaly detection of heterogeneous categorical events and PU learning for anomaly detection by deep learning.

5.2.1 anomaly detection of heterogeneous categorical events

Our framework takes the lead in applying Positive and Unlabeled Learning for anomaly detection. However, the framework can only be applied when feature vectors are extracted. In many real cases of anomaly detection problem, e.g. intrusion detection, insider threat detection, the anomaly behaviors are involved with entities co-occurrence or relations in each event. Thus, we plan to apply PU learning to anomaly detection for heterogeneous categorical events (Fig. 5.1). A probabilistic model will identify the anomaly events where the probability of entities co-occurrence is measured as the anomaly score.

5.2.2 PU learning for anomaly detection by deep learning

Different techniques have been applied to PU learning. However, deep learning can be hardly applied to PU learning because neural networks are supervised classification method and objective function is required for the training. We plan to represent PU learning objective function in a differentiable manner so that deep learning can be applied to PU learning settings.

Rights Statement

©J. Zhang, Z. Wang, J. Yuan, Y.-P. Tan | ACM 2017. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in Proceedings of the 2017 ACM on Multimedia Conference, <http://dx.doi.org/10.1145/3123266.3123304>.

List of Publications

- [1] **Jiaqi Zhang**, Zhenzhen Wang, Junsong Yuan, Tap-Peng Tan “Positive and Unlabeled Learning for Anomaly Detection with Multi-features,” accepted in *ACM MM Conference*, 2017.

- [2] **Jiaqi Zhang**, Zhenzhen Wang, Jingjing Meng, Tap-Peng Tan, Junsong Yuan “Boosting Positive and Unlabeled Learning for Anomaly Detection with Multi-features,” submitted to *IEEE Transactions on Multimedia*.

Bibliography

- [1] B. Liu, W. S. Lee, P. S. Yu, and X. Li, “Partially supervised classification of text documents,” in *ICML International Conference on Machine Learning*, vol. 2. Citeseer, 2002, pp. 387–394.
- [2] 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.
- [3] Y. Ren, D. Ji, and H. Zhang, “Positive unlabeled learning for deceptive reviews detection.” in *EMNLP Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 488–498.
- [4] 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.
- [5] M. N. Nguyen, X.-L. Li, and S.-K. Ng, “Positive unlabeled learning for time series classification,” in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 11, 2011, pp. 1421–1426.
- [6] 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.

- [7] 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.
- [8] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] S. Mascaro, A. E. Nicholso, and K. B. Korb, “Anomaly detection in vessel tracks using bayesian networks,” *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 84–98, 2014.
- [13] 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.

Bibliography

- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] M. Jones, D. Nikovski, M. Imamura, and T. Hirata, “Anomaly detection in real-valued multidimensional time series,” in *International Conference on Big-data/Socialcom/Cybersecurity. Stanford University, ASE*, 2014.
- [19] N. Günnemann, S. Günnemann, and C. Faloutsos, “Robust multivariate autoregression for anomaly detection in dynamic product ratings,” in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 361–372.
- [20] D. Koutra, E. E. Papalexakis, and C. Faloutsos, “Tensorsplat: Spotting latent anomalies in time,” in *Informatics (PCI), 2012 16th Panhellenic Conference on*. IEEE, 2012, pp. 144–149.
- [21] A. Nandi, A. Mandal, S. Atreja, G. B. Dasgupta, and S. Bhattacharya, “Anomaly detection using program control flow graph mining from execution logs,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 215–224.

- [22] I. Melnyk, A. Banerjee, B. Matthews, and N. Oza, “Semi-markov switching vector autoregressive model-based anomaly detection in aviation systems,” *arXiv preprint arXiv:1602.06550*, 2016.
- [23] E. A. Manzoor, S. Momeni, V. N. Venkatakrishnan, and L. Akoglu, “Fast memory-efficient anomaly detection in streaming heterogeneous graphs,” *arXiv preprint arXiv:1602.04844*, 2016.
- [24] M. Amer, M. Goldstein, and S. Abdennadher, “Enhancing one-class support vector machines for unsupervised anomaly detection,” in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*. ACM, 2013, pp. 8–15.
- [25] A. M. Bartkowiak, “Anomaly, novelty, one-class classification: a comprehensive introduction,” *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, no. 1, pp. 61–71, 2011.
- [26] C. Li, Y. Zhang, and X. Li, “Ocvfdt: one-class very fast decision tree for one-class classification of data streams,” in *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data*. ACM, 2009, pp. 79–86.
- [27] S. S. Khan and M. G. Madden, “One-class classification: taxonomy of study and review of techniques,” *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014.
- [28] Y. Song, Z. Wen, C.-Y. Lin, and R. Davis, “One-class conditional random fields for sequential anomaly detection.” in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, 2013, pp. 1685–1691.
- [29] X.-S. Xu, Y. Jiang, X. Xue, and Z.-H. Zhou, “Semi-supervised multi-instance multi-label learning for video annotation task,” in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 737–740.

- [30] K. Zhao, W. Liu, and J. Liu, “Optimal semi-supervised metric learning for image retrieval,” in *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012, pp. 893–896.
- [31] A. Kumar and B. Raj, “Audio event detection using weakly labeled data,” in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1038–1047.
- [32] X. Li and B. Liu, “Learning to classify texts using positive and unlabeled data,” in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 3, no. 2003, 2003, pp. 587–592.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] 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.
- [37] W. Li, Q. Guo, and C. Elkan, “A positive and unlabeled learning algorithm for one-class classification of remote-sensing data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 2, pp. 717–725, 2011.
- [38] W. Yang, X. Yin, H. Song, Y. Liu, and X. Xu, “Extraction of built-up areas from fully polarimetric sar imagery via pu learning,” *IEEE Journal of Selected*

- Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 4, pp. 1207–1216, 2014.
- [39] D. Zhou and B. Schölkopf, “Learning from labeled and unlabeled data using random walks,” *Lecture notes in computer science*, pp. 237–244, 2004.
- [40] S. Chang, Y. Zhang, J. Tang, D. Yin, Y. Chang, M. A. Hasegawa-Johnson, and T. S. Huang, “Positive-unlabeled learning in streaming networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 755–764.
- [41] F. Mordet and J.-P. Vert, “A bagging svm to learn from positive and unlabeled examples,” *Pattern Recognition Letters*, vol. 37, pp. 201–209, 2014.
- [42] 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.
- [43] 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 Proceedings-International Joint Conference on Artificial Intelligence*, 2013, pp. 2176–2182.
- [44] 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.
- [45] C. Elkan and K. Noto, “Learning classifiers from only positive and unlabeled data,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 213–220.
- [46] X.-L. Li, P. S. Yu, B. Liu, and S.-K. Ng, “Positive unlabeled learning for data stream classification,” in *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 2009, pp. 259–270.

Bibliography

- [47] G. Niu, M. C. du Plessis, T. Sakai, and M. Sugiyama, “Theoretical comparisons of learning from positive-negative, positive-unlabeled, and negative-unlabeled data.” *arXiv preprint arXiv:1603.03130*, 2016.
- [48] 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.
- [49] M. Christoffel, G. Niu, and M. Sugiyama, “Class-prior estimation for learning from positive and unlabeled data,” in *Asian Conference on Machine Learning*, 2016, pp. 221–236.
- [50] Z. Tu, J. Cao, Y. Li, and B. Li, “Msr-cnn: applying motion salient region based descriptors for action recognition,” in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 3524–3529.
- [51] 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.
- [52] 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.
- [53] 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.
- [54] 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.