Risky Host Detection with Bias Reduced Semi-Supervised Learning

Shuning Wu
Splunk Corp.
San Jose, CA, USA
shuningw@splunk.com

Joel Fulton
Splunk Corp.
San Jose, CA, USA
jfulton@splunk.com

Ningwei Liu
Splunk Corp.
San Jose, CA, USA
nliu@splunk.com

Charles Feng†
Splunk Corp.
San Jose, CA, USA
wangyanf@splunk.com

Ligang Zhang
School of Natural Resources
University of Nebraska-Lincoln
Lincoln, NE, USA

ABSTRACT

To ensure the cyber security of an enterprise, a SIEM (Security Information and Event Management) system is in place to flag alerts and assign each of them a severity score based on some pre-determined rules. Analysts in the security operations center investigate the high severity alerts to decide if those alerts are truly malicious or not. However, generally the number of alerts is overwhelmingly large, far exceeding the SOC's capacity to handle them, and the majority of them are false positive. There is a great need for a machine learning system to accurately detect the risky hosts. Traditional supervised learning algorithms cannot be directly applied to this problem as very few risky hosts (positive labels) are identified and the positive labels are biased because the SOC analysts only investigate high severity alerts. In this paper, we propose a new distance-based *PU* learning approach, in which we use four different distances to measure similarity to the positive labels and a Gaussian Copula function to capture their correlation structure and ensemble four different distance measures into one joint probability density that we can directly use to infer new labels. The new approach has the advantage of significantly reducing the bias of the inferred labels while traditional supervised *PU* learning increases bias. To quantify bias, we also propose a new bias estimate method. We apply the new bias-reduction Positive Unlabeled (*PU*) learning system to detect host risk in cyber security. Results on real enterprise data indicate that the proposed *PU* learning is able to detect risky hosts effectively while at the same time greatly reducing the label bias. *t*-SNE 2-dimensional visualization also demonstrates that the labels from distance-based *PU* learning are more evenly distributed with higher Kozachenko-Leonenko entropy.

AICS 2019, July 12–13, 2019, Wuhan, Hubei, China

ACM ISBN 978-1-4503-7150-6/19/07…$15.00
https://doi.org/10.1145/3349341.3349365

CCS CONCEPTS

• **Networks~Network security** • Computing methodologies~Machine learning • Information systems

KEYWORDS

PU learning; semi-supervised machine learning; label bias; risky host detection; cyber security

1 Introduction

In this paper, we try to address a machine learning problem where the given dataset has the following three characteristics:

- Dataset only has one class label
- Only a very small percent of the data has labels
- Labels in the dataset are strongly "biased"

For example, risky host detection in cybersecurity is such a problem. The positive samples in the data do not match the true label distribution. The reasons are described below:

- The severity setting may be inaccurate, or biased towards certain use cases
- A majority of alerts have not been investigated. SOC analysts typically skip alerts with low severities, and even some alerts with high priorities may be left alone due to budget and resource limitations. Therefore, we could potentially miss some compromised hosts
- A third challenge to the application of machine learning to such problems is that the dataset is highly unbalanced. In a real enterprise environment, very few hosts (1~2%) in the dataset are labelled.

Our goal, however, is to use the dataset described above and build a machine learning system with Positive Unlabeled (PU) learning to assist the investigation by detecting/ranking risky machines and accelerating the steps to find compromised machines. Traditional supervised machine learning algorithms do not fit our problem as we only have one class label and do not have negative samples (i.e., normal, or not compromised machines). Considering that most machine learning algorithms do not work well for unbalanced datasets, we perform label

propagation as the first step to infer more positive labels. Then we mark the original positive label and the inferred label as "1", and the remaining unlabeled samples as "0". This becomes a traditional binary classification problem and different classifiers can be applied.

Label bias can influence the statistical significance of model parameters, or produce distorted and flawed results. This issue is more prominent for advanced machine learning algorithms such as Support Vector Machines and Neural Networks, where the models tend to capture the high-order non-linear effects from the biased labels. The more an algorithm tries to fit the biased labels, the more severe the bias will be. Therefore, a model's generalizability to the non-biased samples of the population will be deteriorated. A consequence of this is that machine learning systems based on biased labels can lead SOC analysts to investigate only those incidents similar to current high severity cases. As the machine learning model gets retrained from SOC analysts' investigations, the bias will become worse and worse. Of course, we can randomly select a large sample of alerts and ask the SOC analysts to investigate all of them. This will eliminate the label bias but is infeasible in practice due to high labor cost and limited resources.

In summary, our research aims at two challenges:

- very limited availability of labeled data for training of models, i.e., we only have hosts that have been identified as risky, and
- the presence of strongly biased labels, i.e., the analysts only look at high severity alerts

To solve this challenging problem, we propose a distance-based PU learning method for label propagation. This approach reduces label bias compared with traditional supervised label propagation methods. The main contributions of this paper are as follows:

- Propose a new algorithm to quantify the label bias using the *K*-nearest neighbor (KNN) graph method
- Propose a novel distance-based *PU* learning method to reduce label bias. This method integrates multiple distance measures using a Copula function
- Demonstrate a new application of *PU* learning to real industry data. The system is able to effectively detect risky hosts with traditional classification algorithms while at the same time reducing the bias

The remainder of the paper is organized as follows. In Section 2, the related work is summarized. Section 3 explains our proposed algorithm on label bias estimation. Section 4 explains different PU learning techniques for label propagation including distance-based PU learning, which is the core contribution of our paper. Section 5 describes the raw dataset. In Section 6, experimental results on real industry data are discussed, including the comparisons on model bias and performance between contemporary supervised PU learning and our proposed distance-based PU learning methods. Section 7 concludes the whole paper with future work.

2 Related Work

2.1 Label Propagation and Positive Unlabeled (*PU*) Learning

Label propagation is the method of inferring labels on unlabeled samples U from known positive samples P so that machine learning algorithms can be trained on a more balanced dataset. Generally speaking, previous research on learning from Positive and Unlabeled samples can be divided into two categories: supervised ensemble learning and semi-supervised graph inference.

Supervised PU learning [1] generally follows a two-step strategy: Step 1 identifies a reliable positive and/or negative dataset from unlabeled samples, then in Step 2, a series of classifiers are trained on the "reliable" labels and combined to yield the predictions. Such approaches include Roc-SVM [1], S-EM [2], PEBL [2], Adaptive Classifiers [3] and Ensemble SVM [4]. Supervised PU learning has been applied to text/webpage classification [1][2], time series classification [5] and disease gene identification [6].

Semi-supervised label propagation methods are built on a graph Laplacian similarity matrix [7][8][9]. The main idea of these algorithms is that close data points should have similar labels. The algorithms construct a probabilistic transition matrix T based on the graph similarity value $S_{i,j}$ between samples x_i and x_j, that is, $T_{i,j} = P(j \to i) = S_{i,j}/(\sum S_{i,j})$, and keep propagating the labels until the propagated labels converge (Original labels do not change or are less likely to change during the propagation). But these algorithms require both positive and negative labels to start with, which cannot be directly applied to our problem where only risky or "positive" hosts are available.

To the best of our knowledge, we believe this is the first paper to measure the bias in label propagation and propose a distance-based semi-supervised PU learning method to solve this problem.

2.2 Sample Selection Bias

The sample selection bias problem is defined in Cortes's paper [10]. Let X = {xi, i = 1, …, n} denotes the training samples and Y the corresponding labels, and let D denote the true distribution over X × Y. In the sample selection bias problem, some samples z = (x, y) from D are not available to the classification algorithm, hence the algorithm only sees a biased labeled sample P of m labeled points z1, …, zm drawn from a biased distribution D' over X × Y. The sample selection bias can be represented by a random variable s taking values in {0, 1}, where s = 1 represents the sample is selected, otherwise it is not. Using Bayes formula, Cortes [10] defined the bias as below:

$$Pr_D[z] = \frac{\Pr[s=1]}{\Pr[s=1|x]} Pr_{D'}[z] \quad (1)$$

where $Pr_D[z]$ is the probability of drawing z according to the true but unobserved distribution D, and $Pr_{D'}[z]$ is the probability of drawing z according to the biased but observed distribution D'.

3 Label Bias Estimation

One of the key concerns in this paper is how to quantify the label bias. In our case, the SOC analysts never look at the alerts with relatively lower severities, which is the majority of the data, hence the original labels from their investigations are highly biased towards high-severity alerts. Similar to sample selection bias, the label bias can be represented by a random variable l in {0, 1}, where l = 1 represents the sample is labeled. Inspired by Equation (1), we define the overall label bias as:

$$Bias = \sum_i \frac{\Pr[l=1]}{\Pr[l=1|x_i]} \quad (2)$$

$\Pr[l = 1]$ can be estimated by the ratio of the size of positive labeled samples ($|P|$) over the size of overall samples ($|P| + |U|$):

$$\Pr[l = 1] \approx |P|/(|P| + |U|) \quad (3)$$

Clearly $\Pr[l = 1]$ is a constant independent of the data sample x. To estimate $\Pr[l = 1|x_i]$, we adopt an idea from the K-Nearest Neighboring (KNN) affinity graph, which describes the local affinity structure around each training sample x_i. Similarity between training samples is defined by an affinity graph G=(V, E, S), where V is the vertex set including all training samples x_i being one vertex in the graph, and $E = \{e_{i,j}: x_i \leftrightarrow x_j\}$ is the edge set between vertexes with similarity $S = \{S_{i,j}\}$. $S_{i,j}$ is defined as:

$$S_{i,j} = \begin{cases} \exp(-\frac{||x_i - x_j||_2}{2\sigma^2}), x_i \in N_k(x_j), x_j \in N_k(x_i) \\ 0, \quad \text{otherwise} \end{cases} \quad (4)$$

where $N_k(x_i)$ denotes x_i's K-nearest neighbors. With KNN affinity graph, $\Pr[l = 1|x_i]$ can be estimated by counting the positive samples located in x_i's K-nearest neighbors, k_i, for a certain threshold of S_T such that $S_{i,j} \geq S_T$, that is,

$$\Pr[l = 1|x_i] \approx \frac{\sum_p \mathbf{1}_{N_k(x_i)}(x_p)}{k_i} \quad (5)$$

where $\mathbf{1}_{N_k(x_i)}(x_p) := \begin{cases} 1, & if\ x_p \in P \cap N_k(x_i) \\ 0, & \text{otherwise} \end{cases}$

Equations (2), (3) and (5) provide us a quantitative way to estimate the label bias in a classification problem. The algorithm is summarized below:

Algorithm: Label Bias Estimation

Input:

Sample matrix $\boldsymbol{X} \in \boldsymbol{R^{n \times q}}$; Sample ID set $\{\boldsymbol{j}|\boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{n}\}$;

Positive samples *P*; KNN graph parameters $\boldsymbol{\sigma}$ and $\boldsymbol{S_T}$

1. Construct the *K*-nearest neighbor graph and affinity matrix *S*
2. Calculate $\mathbf{Pr}[\boldsymbol{l} = \boldsymbol{1}] \approx \frac{|\mathbf{P}|}{\boldsymbol{n}} = \boldsymbol{p_0}$
3. Initialize *i*=1, $\boldsymbol{b} \in \boldsymbol{R^{n \times 1}}$ and set *b* as a zero vector
4. Construct positive sample ID set: $\boldsymbol{P_{idx}} = \{\boldsymbol{j}|\boldsymbol{x_j} \in \boldsymbol{P}\}$
5. Repeat: For *i*=1 to *n*
   a. Construct $\boldsymbol{x_i}$'s *k*-nearest neighbor ID set for given $\boldsymbol{S_T}$: $\boldsymbol{K_{idx}} = \{\boldsymbol{j}|\boldsymbol{x_j} \in \boldsymbol{N_k(x_i)}\ \boldsymbol{and}\ \boldsymbol{S_{i,j}} \geq \boldsymbol{S_T}\}$
   b. Count interception of set $\boldsymbol{P_{idx}}$ and set $\boldsymbol{K_{idx}}$: $\boldsymbol{c_i} = \mathbf{max}(|\boldsymbol{P_{idx}} \cap \boldsymbol{K_{idx}}|, \boldsymbol{1})$
   c. Calculate the bias for sample $\boldsymbol{x_i}$: $\boldsymbol{b_i} = \boldsymbol{p_0} \times \frac{|\boldsymbol{K_{idx}}|}{\boldsymbol{c_i}}$

Output:

Sample bias estimation *b*; Overall bias estimation $\sum \boldsymbol{b_i}$

4 *Pu* Learning & Label Propagation

As mentioned in the introduction, we have very few hosts (about 1%) with positive labels and many classification models do not work well for highly unbalanced datasets. Label propagation is desirable to infer more positive labels from the unlabeled samples before applying any classification algorithm. In this section, we will elaborate on the supervised PU learning approach and our new distance-based PU learning approach.

4.1 Supervised *PU* Learning

The label propagation problem can be viewed as a supervised Positive Unlabeled Learning problem [1], where the data only includes positive and unlabeled samples, but no negative samples. We use P to denote the positive samples (i.e. risky hosts from analysts' annotations) and U to denote unlabeled samples. The supervised PU learning is performed as follows:

1. Assign each sample in *P* the class label 1;
2. Assign each sample in *U* the class label 0;
3. Build a classifier using *P* and *U*;
4. Apply the classifier to classify the samples from *P* and *U*. Select the top 10% samples with highest predictions and label them as 1. The 10% cutoff is fixed across different approaches so we can compare the performance later from different approaches.

Any classifier can be used in Steps 3 and 4. We chose the Support Vector Machine (SVM) in our experiment as this method is widely used in other PU learning research.

4.2 Distance-based *PU* Learning

We suspect the supervised PU learning approach will increase label bias and this hypothesis will be verified in the experiments. In contrast to using a classifier for label propagation, we propose a new distance-based PU learning approach, in which we use distance to measure similarity to the existing labels. In order to capture different characteristics of similarities, we use four different distance measures: Manhattan, Canberra, Euclidean, and Mahalanobis distance. Moreover, we use the Copula function to capture the correlation structure between different distances and ensemble four different similarity measures into one joint probability density that we can use to infer new labels. The process is described as follows:

1. Calculate the centroid of positive samples Pc: $P_c = (\sum_{i \in P} x_i)/|P|$, where $|P|$ is the size of positive samples

2. Calculate different distances between sample $x_j, j \in (P \cup U)$ and centroid P_c, including:

Manhattan distance: Manhattan distance is the sum of the absolute values of the horizontal and the vertical distance,

$d_{man} = \sum_{j\in(P\cup U)} \|x_j - P_c\|_1$, where $\|x_j - P_c\|_1$ denotes L_1 norm of $(x_j - P_c)$

Euclidean distance: Euclidean distance is the straight-line distance between two points in Euclidean space. It is the most fundamental and widely used distance measure.

$$d_{euc} = \sqrt{\sum_{j\in(P\cup U)} \|x_j - P_c\|_2},$$

where $\|x_j - P_c\|_2$ denotes L_2 norm of $(x_j - P_c)$

Mahalanobis distance: Mahalanobis distance measures the distance between a point and a distribution. This distance increases as the point moves away from the mean of the distribution.

$$d_{mah} = \sqrt{\sum_{j\in(P\cup U)} (x_j - P_c)^T S^{-1} (x_j - P_c)},$$

where S^{-1} is the correlation matrix on x

3. Different distances measure different similarities to the positive samples with different distributions, and are correlated. We want to build a multivariate model from each distance's marginal distribution. This model not only captures the dependency structure among different distances but also combines different distance similarities into one single similarity score. The Copula model is a good fit for this purpose. It provides a means of inference after modeling a multivariate joint probability from different distance measures. A copula $C(u_1, \dots, u_m;\ \theta)$ is a joint cumulative distribution function over m continuous random variables, each with a uniformly distributed marginal between [0, 1]. The joint probability density function can be obtained by taking the mth derivative of $F(d_1, \dots, d_m)$:

$$f(d_1, \dots, d_m) = \prod_{i=1}^{m} f_i(d_i) \cdot c(F_1(d_1) \dots F_m(d_m);\ \theta)$$

where $c(.)$ is the copula density. For the Gaussian Copula, the parameter θ can be estimated with the maximum log-likelihood method [11]:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum log \ \{ \prod_{i=1}^{m} f_i(d_i) \cdot c(F_1(d_1) \dots F_m(d_m);\ \theta)\}$$

We choose the Gaussian Copula because the model is easy to implement and more importantly, it works well for our risky host detection problem. We also tried t-student copula but there was no added benefit from the more complex copula structure.

4. Output the probability density for each sample from the Gaussian Copula distribution, select the top 10% samples with lowest probability density and label them as 1, as the samples within a low-density region of the joint distribution are more likely to be anomalies [12].

5 Raw Data & Feature Engineering

Our dataset is one month of security events with 25,715 distinct hosts and 100 features. And 1.17% of the hosts have been verified as compromised by security analysts. Then we perform the label propagation and combine verified labels with the inferred ones as the targets for the machine learning model. The final modeling dataset looks like this:

Table 1: Example of Final Modeling Dataset

Host ID	Summary feature 1	Indicator feature 2	Temporal feature 3	…	**Label**
Host1	13	1	0.65	…	1 (risky-**original**)
Host2	25	0	2.74	…	1 (risky-**inferred**)
Host3	4	0	1.33	…	0 (unlabeled)

6 Experiment Results

The settings of the experiment are:

- K-Nearest Neighboring (KNN) affinity graph: σ is a constant in $S_{i,j}$ independent of x_i or x_j, so its value is trivial and won't influence the ranking of nearest neighbors. In practical calculations, σ should be set relatively small as large σ will cause similarities to vanish too quickly, therefore leaving too many neighbors with the same rankings. In our experiment, we set $\sigma = 30$. Similarity threshold S_T will be varied in the experiment.
- Classifier used in Supervised PU learning: Support Vector Machine (with radial basis function kernel)
- Classifier used in final prediction: Logistic Regression (Generalized linear model with Binomial family and Logit link function)
- As a common practice, we split the data randomly into training (75% of the samples) and test (remaining 25%) sets. We use area under the curve (AUC) as a way to measure performances of different classification models.

6.1 Label Propagation Bias and Visualization

The first thing we show here is the label bias from different label propagation techniques. As a comparison, we also calculate the bias from the initial labels generated from SOC analysts' investigations. Table 3 lists the estimated label bias with different S_T values, ranging from 0.75 to 0.90. It can be seen that as S_T drops, the estimated label bias decreases monotonically as well. This is expected as $S_T \to 0$, the overall bias will be approaching its minimum value $\sum b_i \to \sum 1 = n$.

Table 2: Estimated Label Bias over Different S_T

S_T	Initial Label	Supervised *PU* Learning	Distance-based *PU* Learning
0.90	35,187.24	35,997.06	34,875.30
0.85	35,068.87	35,767.16	34,842.73
0.80	35,002.75	35,569.35	34,811.42
0.75	34,889.02	35,409.58	34,790.61

Furthermore, we notice that supervised PU learning has the highest estimated bias over different S_T values, while distance-based PU learning has the lowest bias. More importantly, the bias from distance-based PU learning is consistently lower than the bias from the initial labels while this is opposite for supervised PU learning. This implies that distance-based PU learning reduces label bias globally, that is, the label bias is consistently reduced with broader neighborhoods (i.e., smaller S_T). Supervised PU learning, on the other hand, amplifies the bias. Therefore, our hypothesis that distance-based PU learning helps to reduce label bias is supported from the results in Table 2.

Next, we use t-Distributed Stochastic Neighbor Embedding (t-SNE) [13]. t-SNE is a nonlinear dimensionality reduction technique for embedding high-dimensional data into low dimensions (i.e., 2 dimensions). It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. Once the feature space dimension is reduced, we can create 2-dimensional scatter plots on the labelled samples.

In addition to visualizations, we use the Kozachenko-Leonenko non-parametric k-nearest neighbor entropy estimator [14] to measure the distribution diversity of inferred labels over the reduced 2-dimentional feature space from t-SNE embedding. This approach estimates the entropy of the underlying distribution from the Euclidean distance between each sample and its k-nearest neighbors. The entropy estimate on sample X is defined as:

$$\hat{H}(X) = \psi(n) - \psi(k) + \log(c_d) + \frac{d}{n}\sum_{i=1}^{n}\log(\varepsilon(i)) \quad (6)$$

where n is the number of samples, d is the number of dimensions (here d=2), ψ is the digamma function which is given by the logarithmic derivative of the Gamma function Γ,

$$\psi(z) \equiv \frac{d}{dz} ln\Gamma(z) = \frac{\Gamma'(z)}{\Gamma(z)} \quad (7)$$

c_d is the volume of the d-dimensional unit-ball centered at a sample point x_i (i.e., $||x - x_i|| < 1$), for Euclidean distance, c_d is given by:

$$c_d = \pi^{\frac{d}{2}}/\Gamma(1 + d/2) \quad (8)$$

and $\varepsilon(i)$ is the distance of the i^{th} sample x_i to its k^{th} nearest neighbor. With Equations (6) to (8), we can calculate the Kozachenko-Leonenko entropy from *t*-SNE embedding. Higher entropy implies that the labeled samples in *t*-SNE 2-dimensional space are more evenly distributed.

The scatter plots in Figures 1 and 2 display the results from supervised PU learning and distance-based PU learning respectively. The x and y axes are taken from the two lower dimensional components from t-SNE embedding. In both Figures 1 and 2, the red triangles represent initial labels and the blue crosses represent the inferred labels. It may seem that Figure 1 has fewer blue crosses then Figure 2, but they actually have the same number of inferred labels. The reason is that Figure 1 has overlapped crosses in certain regions, while the blue crosses are more evenly distributed in Figure 2. We want to emphasize that both the visualizations and Kozachenko-Leonenko entropy values are consistent with the findings from our label bias estimates, which we will discuss further below.

Supervised PU learning fits the SVM model on the initial biased labels. It can be seen from Figure 1 that the blue inferred labels cover the red initial labels very well, but the blue points tend to concentrate in certain regions, implying that the propagated labels are possibly chasing some influential initial labels, therefore the original label bias is "amplified" after label propagation. The Kozachenko-Leonenko estimator with k=200 yields an entropy value of 1.88.

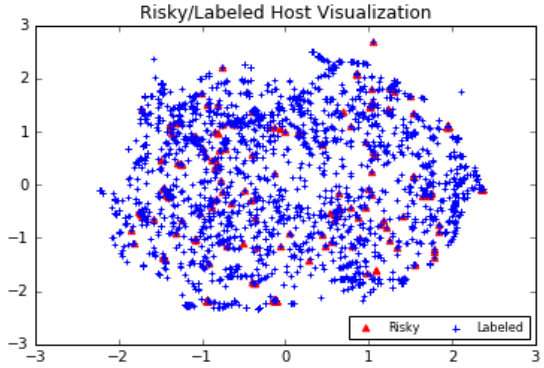

Figure 1: Label distribution from supervised *PU* learning

Distance-based PU learning, on the other hand, does not rely on initial labels as much as supervised PU learning. It only uses the centroid of positive labels as a reference point and mainly applies unsupervised distance measures and the Copula function in the training process to assign the labels. In contrast to the supervised PU learning, the blue inferred labels are more evenly distributed while at the same time they still cover the initial labels pretty well, hence distance-based PU learning is not only capable of incorporating information from the initial positive labels, but also "smoothing" the label selection bias. The Kozachenko-Leonenko estimator yields an entropy value of 2.85, which is 1.5 times higher than the estimated entropy from supervised PU learning, implying that the labeled points are more evenly distributed in t-SNE 2-dimensional space as shown in Figure 2.

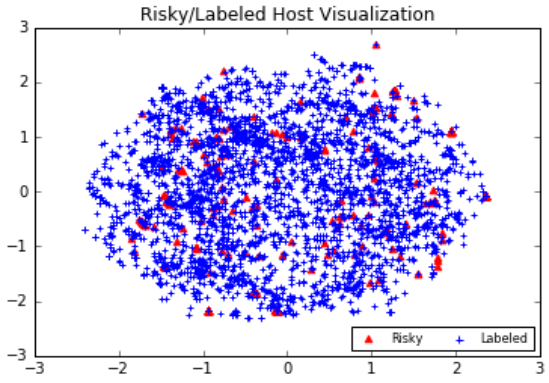

Figure 2: Label distribution from distance-based *PU* learning

6.2 Classification Performance

In this section, we will go over the classification results from supervised PU learning and our distance-based label propagation method. As the main focus of this paper is on finding a metric and algorithm to measure label bias and to develop a new method that can reduce label bias and the machine learning algorithm itself is not our main concern. Hence, we chose to use a simple logistic regression classifier to predict the likelihood of a host being compromised. Table 3 shows the AUC values on test data from different label propagation methods. As expected, supervised PU learning obtains the highest AUC value as it goes over the training data twice: first it uses SVM to fit the initial label for label propagation and second it uses logistic regression for the final prediction. It is promising that our new distanced-based semi-supervised learning method - in which only the centroids of the initially verified positive samples were used for label propagation - achieves a high AUC of 0.87, and reduces the label bias at the same time.

Table 3: AUC Values on Test Data

	Area Under Curve (AUC)
Initial Label	0.8394
Supervised *PU* Learning	0.9639
Distance-based *PU* Learning	0.8711

Table 4 lists the detection rates on the top 5% to 20% of predictions respectively. Considering that we only have 1.17% of risky hosts in the test dataset, it is promising that simple Logistic Regression is able to detect 85% of the true risky cases with the top 20% of predictions from distance-based PU learning. As expected, the classifier from supervised PU learning obtains the highest detection rate over different portions of predictions.

Table 4: Detection Rates on Top 5%~20% Predictions

Top % of Predictions	Initial Label	Supervised *PU* Learning	Distance *PU* Learning
5%	68.33	76.67	**60.00**
10%	68.33	90.00	**75.00**
15%	68.33	93.33	**78.33**
20%	70.00	96.67	**85.00**

7 Conclusions and Discussions

We propose a new distance-based PU learning approach, in which we use four different distances to measure similarity to the existing labels and the Gaussian Copula function to capture their correlation structure and ensemble four different similarity measures into one joint probability density that we can directly use to infer new labels. The new approach has the advantage of significantly reducing the bias of the inferred labels while contemporary supervised PU learning increases bias in comparison, based on the new bias estimate method presented in the paper. The bias estimate is consistent with the visualizations from t-SNE embedding method and Kozachenko-Leonenko entropy calculation. This indirectly validates our bias estimate method. Our new bias estimate method and new distance-based PU learning method can be applied to a wide range of real-life scenarios where only limited one-class labels are available and they are strongly biased, even though the data and the application illustrated in this paper are in the cyber security domain.

From the comparison of classification results in risky host detection, logistic regression with distance-based PU learning achieves a good balance between label bias and model performance. It significantly reduces the label bias while at the same time still producing satisfactory classification performance. Supervised PU learning, on the other hand, fits the current biased labels very well with the best classification results, but it amplifies the label bias, which implies that its performance may deteriorate on true unbiased labels.

As to future work, we will do more theoretical research to explore what kinds of problems will be best suitable for distance-based PU learning. From a practical application perspective, we will work on optimizing the classification performance after the label propagation by using more sophisticated classifiers, other than logistic regression, as described in this paper.

REFERENCES

[1] B. Liu, Y. Dai, X. Li, W. S. Lee and P. S. Yu. "Building text classifiers using positive and unlabeled examples." Proceedings of the Third IEEE International Conference on Data Mining, 2003.

[2] H. Yu, J. Han and K. C. Chang. "PEBL: positive example-based learning for web page classification using SVM." Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, 2002

[3] C. Elkan and K. Noto. "Learning classifiers from only positive and unlabeled data." Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008

[4] M. Claesen, F. D. Smet, J. A. K. Suykens and B. D. Moor. "A Robust Ensemble Approach to Learn from Positive and Unlabeled Data Using SVM Base Models." Neuro-computing: Special Issue on Advances in Learning with Label Noise, 2015

[5] M. N. Nguyen, X. Li and S. K. Ng. "Positive Unlabeled Learning for Time Series Classification." Proceedings of the 22nd international joint conference on Artificial Intelligence, 2011
[6] P. Yang, X. L. Li, J. P. Mei, C. K. Kwoh and S. K. Ng. "Positive-Unlabeled Learning for Disease Gene Identification." Bioinformatics 28 (20): 2640-2647, 2012
[7] O. Delalleau, Y. Bengio and N. L. Roux. "Efficient Non-Parametric Function Induction in Semi-Supervised Learning." Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics, 2005
[8] D. Zhou, O. Bousquet, T. N. Lal, J. Weston and B. Schölkopf. "Learning with local and global consistency." Advances in Neural Information Processing Systems 16: 321-328, 2004.
[9] X. Zhu and Z. Ghahramani. "Learning from labeled and unlabeled data with label propagation." Carnegie Mellon University, 2002
[10] C. Cortes, M. Mohri, M. Riley and A. Rostamizadeh. "Sample Selection Bias Correction Theory." Proceedings of the 19th international conference on Algorithmic Learning Theory, 2008
[11] S. G. Iyengar. "Decision-making with heterogeneous sensors-a copula-based approach." PhD Dissertation, 2011
[12] K. Veeramachaneni, I. Arnaldo, V. Korrapati and C. Bassias. "AI2: Training a big data machine to defend." IEEE International Conference on High Performance and Smart Computing (HPSC), 2016
[13] V. Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research 9:2579-2605, 2008
[14] D. Lombardi and S. Pant. "A non-parametric k-nearest neighbor entropy estimator." Physical Review E, Volume 93, Issue 1, 2016