Fraud Detection under Multi-Sourced Extremely Noisy Annotations

Chuang Zhang[1*], Qizhou Wang[2*], Tengfei Liu[3], Xun Lu[3], Jin Hong[3], Bo Han[2], Chen Gong[1†]
[1]PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]Hong Kong Baptist University, [3]Ant Group
chen.gong@njust.edu.cn

ABSTRACT

Fraud detection in e-commerce, which is critical to protecting the capital safety of users and financial corporations, aims at determining whether an online transaction or other activity is fraudulent or not. This problem has been previously addressed by various fully supervised learning methods. However, the true labels for training a supervised fraud detection model are difficult to collect in many real-world cases. To circumvent this issue, a series of automatic annotation techniques are employed instead in generating multiple noisy annotations for each unknown activity. In order to utilize these low-quality, multi-sourced annotations in achieving reliable detection results, we propose an iterative two-staged fraud detection framework with multi-sourced extremely noisy annotations. In *label aggregation stage*, multi-sourced labels are integrated by voting with adaptive weights; and in *label correction stage*, the correctness of the aggregated labels are properly estimated with the help of a handful of exactly labeled data and the results are used to train a robust fraud detector. These two stages benefit from each other, and the iterative executions lead to steadily improved detection results. Therefore, our method is termed "Label Aggregation and Correction" (LAC). Experimentally, we collect millions of transaction records from Alipay in two different fraud detection scenarios, *i.e.*, credit card theft and promotion abuse fraud. When compared with state-of-the-art counterparts, our method can achieve at least 0.019 and 0.117 improvements in terms of average AUC on the two collected datasets, which clearly demonstrate the effectiveness.

CCS CONCEPTS

• **Information systems** → **Electronic commerce**.

KEYWORDS

Fraud detection, label noise, crowdsourcing.

ACM Reference Format:
Chuang Zhang, Qizhou Wang, Tengfei Liu, Xun Lu, Jin Hong, Bo Han, Chen Gong. 2021. Fraud Detection under Multi-Sourced Extremely Noisy Annotations. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482433

∗ Contribute equally to the work.
† Corresponding author.

CIKM '21, November 1–5, 2021, Virtual Event, Australia.

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482433

1 INTRODUCTION

The problem of fraud detection has a large impact in the field of e-commerce and has attracted a great deal of attention by e-commerce platforms such as Alipay. There are many types of fraud on e-commerce platforms, such as account takeover fraud, credit card theft, promotion abuse fraud, *etc.* Account takeover means fraudsters use stolen credentials to access a genuine account. When it is successful, fraudsters can take control of the account and conduct some illegal activities. Credit card theft means someone's credit card is stolen by others for unauthorized purchases. Promotion abuse fraud refers to fraudsters intentionally create multiple fake accounts to repeatedly enjoy shopping discounts issued by shopping platforms (*e.g.*, Taobao and Tmall). These fraud activities bring about a huge economic loss every year for e-commerce platforms and numerous users. Unfortunately, manually identifying the abnormal activities and fraudsters is infeasible, owing to the massive throughput of the business and the great diversity of fraud behaviors. Therefore, in e-commerce, a high-quality automatic fraud detection system is strongly demanded.

Fraud detection is generally viewed as a binary classification problem in assigning the label of "genuine" or "fraudulent" to an unknown transaction[1]. Given the historical records with accurate annotations, previous methods typically formulate this problem as a general supervised problem [8, 20, 32]. However, the heavy reliance on accurate annotations critically restricts the scenarios of deploying these methods, due to the expensive cost of time and money in labeling. To circumvent this problem, a series of primitive labeling techniques are deployed to automatically generate multiple labels for each transaction with high efficiency (*cf.*, Figure 1). Although the annotation process is quite efficient, the resulting labels are of extremely low quality, since they are simply determined by some fixed business rules or the outputs of out-of-date models in similar business scenarios. As a result, the performance of canonical supervised methods will be largely affected [38]. Besides, we want to mention that the conventional crowdsourcing methods [14, 30, 53] dealing with multi-sourced labels are not applicable here owing to the severe label noise existed in the annotation results. Therefore, it is highly demanded to devise a reliable learning approach with massive multi-sourced noisy annotations for accurate fraud detection in e-commerce.

[1]We use the term "transaction" to refer to fraud activities throughout this paper.

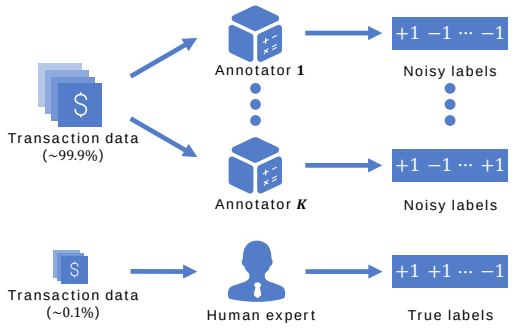

Figure 1: The annotation process of our method. For every transaction, a series of automatic annotation techniques (K annotators) are employed in assigning the label of "genuine" (-1) or "fraudulent" ($+1$). The quality of the assigned labels could be low due to the fixed criteria or business rules, resulting in multi-sourced noisy labels. If necessary, the true label for a small amount ($\sim 0.1\%$) of transaction records can also be provided manually with some acceptable extra costs.

In this paper, we propose an effective learning framework for fraud detection under multi-sourced extremely noisy annotations. Concretely, we construct an iterative learning paradigm that consists of two main stages, namely, *label aggregation stage* and *label correction stage*. Specifically, label aggregation stage tries to infer a reliable label for each transaction from the multi-sourced noisy annotations, for which we adopt a weighted voting scheme that adaptively considers the quality of each annotator in deciding transactions as "genuine" and "fraudulent". Although label aggregation stage is able to convert the multi-sourced labels of a transaction into a single one, the quality of the resulting labels can still be further improved as the correctness of individual transactions has not been properly investigated. Therefore, label correction stage is deployed, where the confidence score of each individual transaction regarding the aggregated label is estimated. Then, the confidence scores are sent to a robust learning framework in constructing the final classifier for fraud detection. It is worth noting that the confidence scores are trustable due to the employment of a small set of transactions with verified true labels[2]. Therefore, the resulting classifier can further enhance the quality of aggregated labels rendered by the previous aggregation stage. These two stages execute alternatively and they collaboratively lead to the increased performance of fraud detection.

We collect more than a million transaction cases in two different real-world fraud scenarios from Alipay, which are related to credit card theft (*CCT*) and promotion abuse fraud (*PAF*), respectively. With multi-sourced noisy labels automatically assigned by the aforementioned annotation process, we construct two corresponding fraud detection datasets. Then, we conduct extensive experiments on the two datasets in comparison with state-of-the-art counterparts, and our method achieves $0.019 \sim 0.188$ improvements on *CCT* and $0.117 \sim 0.172$ improvements on *PAF* in terms of the average AUC on test sets.

[2]Note that the cost for annotating a small amount of transactions is practically acceptable, and this small clean set has been widely used in many prior works such as [5, 13, 46].

2 RELATED WORK

In this section, we provide a comprehensive overview on fraud detection, learning with label noise, and learning from crowds.

2.1 Fraud Detection

Fraud detection has drawn a great deal of attention in the literature of machine learning [1, 3]. It can be viewed as a binary classification problem and aims at distinguishing fraudulent transactions from the genuine ones. In general, previous learning methods can be attributed to three categories, namely, supervised techniques, unsupervised techniques, and semi-supervised techniques.

Supervised techniques rely on the historical transactions with true labels, in which the true labels are collected either from the feedback of users or the investigation results of financial corporations. These methods adapt traditional supervised techniques into fraud detection, such as deep neural network [8, 48] and decision tree [32]. However, the collection of true labels is usually along with unaffordable time and labor costs. In contrast, unsupervised techniques [10, 15, 19] do not require any true label. Instead, they employ various anomaly detectors with the assumption that the outliers are fraudulent. Nonetheless, the reliability of the detection results cannot be guaranteed due to the absence of supervision. Therefore, semi-supervised techniques [9, 47] are further utilized to conduct fraud detection, which can be viewed as a combination of the above two kinds of methods. The methods belonging to this category adopt a large scale of unlabeled transactions with a few correctly labeled ones. Obviously, their deployment costs are lower when compared with the supervised techniques, and the detection reliability is much better than the unsupervised techniques.

Different from previous works, we aim at learning from historical transactions with multi-sourced noisy annotations, which may provide more reliable supervision with the minimum labeling cost in comparison with the completely supervised and unsupervised cases. Similar to semi-supervised techniques, a small portion of transactions with true labels are also used in our method, but differently, large scale transactions with multi-sourced noisy annotations are also utilized here to provide more potential supervision.

2.2 Learning with Noisy Labels

Label noise learning is one of the most representative learning settings in the field of weakly supervised learning [25, 26, 50, 56, 57], which aims at learning robust classifiers in the presence of data with inaccurate labels. According to the generation process of the noisy labels, the studies on label noise learning mainly fall into the following three categories [17], namely, random classification label noise, class conditional label noise, and instance dependent label noise.

The setting of random classification label noise assumes that noisy labels are corrupted completely at random, and various noise-tolerant loss functions have been explored [22, 23, 35, 38, 51]. The works of class conditional label noise additionally assume that noisy labels are dependent on the unknown true labels. It considers the fact that some class pairs are more prone to be mutually mislabeled. The noise transition matrix is adopted to depict such phenomenon, and various statistically consistent learning frameworks are proposed, such as [24, 33, 40, 54]. Instance dependent label noise

setting is much general but scarcely studied due to its high complexity, which assumes that the label corruption is related to data themselves in addition to the unknown class labels. The representative works are [5, 13, 16, 36, 49]. Note that, the setting of label correction stage in our method exactly coincides with learning with noisy labels. We explicitly consider instance dependent label noise in our method, which is the most general case among the three noise types mentioned above. Besides, our method is also feasible in learning with severe label noise, which is a challenging situation where most of the existing robust methods for handling label noise may fail.

2.3 Learning from Crowds

Learning from crowds (*a.k.a.* crowdsourcing) provides a cost-effective solution when the true labels of a dataset are not available [37, 44, 58]. It allows a group of cheap annotators to do the labeling task, and thus each datum is assigned with multiple labels provided by the annotators. Such labels might be noisy as the annotators are not experts and may lack domain knowledge. The ultimate goal is to use these multiple low-quality labels to infer the corresponding true labels of data points.

In the literature of learning from crowds, existing methods can be divided into two major categories depending on how the inference is integrated with the training of the classifier. The first category considers a two-staged learning strategy which first infers the ground-truth labels using multi-sourced labels, and then trains the target classifier based on the inferred labels. Intensive efforts have been devoted to the inferring stage, such as [43, 52, 53]. The second category is referred as the joint approach, which simultaneously infers true labels and trains the classifier [2, 30, 42]. These methods allow the two processes to benefit from each other, and thus they generally perform better than those belonging to the first category. Although the above methods have achieved promising performance on various tasks, they may fail when the quality of annotators is extremely unsatisfactory. Therefore, traditional learning methods for crowdsourcing problem are unsuitable in remedying the severe label noise for our task of fraud detection.

3 OUR METHOD

To begin with, we fix some necessary notions in this paper. We consider a dataset of N transactions $\{x_i\}_{i=1}^N \in \mathbb{R}^d$ with the associated true labels $\{z_i\}_{i=1}^N \in \{-1, +1\}$, where d is the dimension of the feature vector for the transactions. Here, the true labels are not accessible in most cases. Instead, each transaction is equipped with K annotations as shown in Figure 1. Let $\{y_i^k\}_{k=1}^K$ denote the K noisy annotations corresponding to x_i, with $y_i^k \in \{-1, +1\}$ being the assigned label for the i-th transaction given by annotator k, where $y_i^k = -1$ for a genuine transaction and $y_i^k = +1$ for a fraudulent transaction. Note that, the assigned label y_i^k may be different from the corresponding true label z_i, *i.e.,* $y_i^k \neq z_i$. Therefore, our goal is to construct a scoring function $f(x) : \mathbb{R}^d \mapsto \mathbb{R}$ with multi-sourced noisy annotations, such that the corresponding classifier $g(x) = \text{sgn}(f(x))$ could provide an accurate prediction for any new transaction x, where $\text{sgn}(\cdot)$ denotes the sign function that returns $-1/+1$ according to the sign of its input.

3.1 Learning Framework

In this section, we briefly introduce the iterative learning framework for fraud detection (see Figure 2), where the following two stages are deployed, namely, *i. Label aggregation stage*: integrating the multi-sourced noisy annotations into a single one by considering the quality levels of different annotators; *ii. Label correction stage*: learning a reliable classifier by considering the confidence of aggregated labels.

label aggregation stage aggregates the original multi-sourced labels by employing a weighted voting scheme. Therein, each annotator is related to a pair of weights, respectively characterizing its quality in deciding a transaction as genuine and fraudulent. In label correction stage, we further estimate the confidence scores of the aggregated labels of all transactions. The confidence scores can be effectively estimated by canonical clustering techniques, and the results are of high quality with the aid of a small set of transactions with manually verified true labels. The aggregated labels and their confidence scores are further utilized to train a statistically consistent classifier, which could be effectively used for judging a new transaction as fraudulent or not. The two stages proceed iteratively and each of them is benefited from the other, so the final performance of our method can be enhanced. In the next two parts, we will describe label aggregation stage and label correction stage in detail.

3.2 Label Aggregation Stage

In this section, we describe label aggregation stage in the proposed learning framework, which integrates the multi-sourced labels of each transaction into a single one.

3.2.1 Weighted Voting. A straightforward method of label aggregation for multi-sourced annotations is majority voting. Accordingly, the aggregated label y_i for the i-th transaction can easily be obtained by

$$y_i \leftarrow \text{sgn}\left(\sum_{k=1}^K 1^{\mathbb{1}[y_i^k=+1]}(-1)^{\mathbb{1}[y_i^k=-1]}\right), \tag{1}$$

where $\mathbb{1}[\cdot]$ denotes the indicator function that equals to 1 if the condition in the bracket is true and 0 otherwise.

However, the majority voting does not consider the quality of different annotators in assigning genuine and fraudulent labels, which may lead to inferior results. Therefore, in our method, a weighted voting method is adopted in label aggregation, namely

$$y_i \leftarrow \text{sgn}\left(\sum_{k=1}^K v_{k,+1}^{\mathbb{1}[y_i^k=+1]}(-v_{k,-1})^{\mathbb{1}[y_i^k=-1]}\right), \tag{2}$$

where $v_{k,+1}$ and $v_{k,-1}$ are defined as the overall labeling quality of the k-th annotator for providing the positive and negative labels. Intuitively, if the k-th annotator performs better in deciding the fraudulent transactions than determining the genuine ones, the value of $v_{k,+1}$ should be larger than $v_{k,-1}$.

3.2.2 Estimation of $v_{k,+1}$ and $v_{k,-1}$. Note that, the paired weights to depict annotator quality $v_{k,+1}$ and $v_{k,-1}$ should be properly estimated as they play a critical role in inferring the aggregated labels. Here, we cast the estimation of paired weights of annotators $\{(v_{k,+1}, v_{k,-1}) : k = 1, \cdots, K\}$ as a maximum likelihood estimation

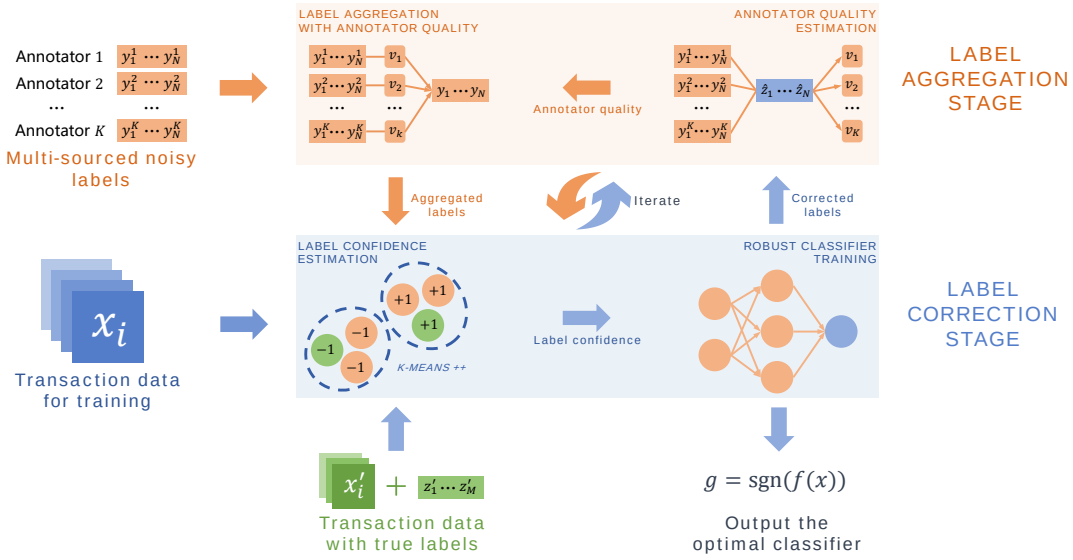

Figure 2: The overview of our framework, where v_i $(i = 1, \cdots, K)$ denotes the annotator quality, y_i and $\hat{z}_i$ $(i = 1, \cdots, N)$ denote the aggregated labels and corrected labels, respectively. The *label aggregation stage* integrates the multi-sourced annotations for every transaction. Therein, the quality of annotators in deciding fraudulent/genuine cases are considered, and a weighted voting strategy is deployed accordingly. Then, with the help of a small amount ($M \ll N$) of verified transaction data, the *label correction stage* further estimates the confidence scores of the aggregated labels for each transaction via canonical clustering techniques (*e.g.*, k-means++ in this work), and a statistically consistent learning method is deployed in training the robust classifier for fraud detection. These two stages execute iteratively until convergence and output is the final robust fraud detector.

problem. Taking the predicted labels $\{\hat{z}_1, \cdots, \hat{z}_N\}$ given by the robust classifier in label correction stage as the pseudo labels[3], we have the following optimization problem:

$$\arg\max_{v_{k,+1}, v_{k,-1}} \sum_{i=1}^{N} \log a_i^{\mathbb{1}[\hat{z}_i=+1]} (1-a_i)^{\mathbb{1}[\hat{z}_i=-1]}, \tag{3}$$

where a_i is defined as the probability $P(y_i = +1|x_i)$ and is calculated by $\sigma\left(\sum_{k=1}^{K} v_{k,+1}^{\mathbb{1}[y_i^k=+1]} \left(-v_{k,-1}\right)^{\mathbb{1}[y_i^k=-1]}\right)$ with $\sigma(\cdot)$ being the *sigmoid* activation function. By converting the estimation of annotator quality into a maximum likelihood estimation algorithm, the traditional gradient descent method can be directly used for optimization.

By combining Eq. (2) and Eq. (3), we have the outputs of label aggregation stage, namely, the aggregated labels $\{y_1, \cdots, y_N\}$ for the N transactions.

3.3 Label Correction Stage

Although the above label aggregation stage is effective in eliminating the noisy annotations, the aggregated labels $\{y_1, \cdots, y_N\}$ are still expected to be purified as the initial multi-sourced labels are severely corrupted. Therefore, we further invoke label correction stage, which focuses on estimating the confidence score of each aggregated label and accordingly learns the target classifier for fraud detection. Note that, the setting of label correction stage exactly coincides with robust learning with noisy labels [38].

[3]The establishment of robust classifier in label correction stage will be introduced in Section 3.3 below.

In this section, we first introduce the general risk-consistent learning framework and define the confidence of aggregated labels. Then, we provide the specific method to estimate these confidence scores which help to yield the final clean labels.

3.3.1 Risk-Consistent Learning. Let X, Z, and Y denote the random variables corresponding to the input feature x, clean label z, and noisy label y, respectively. If the distribution over (X, Z) is provided in advance, we can have the optimal classifier $g^*(f(X))$ for fraud detection by minimizing the following expected risk [7]:

$$R(f) = \mathbb{E}_{(x,z)\sim(X,Z)} \left[\ell(f(x), z)\right], \tag{4}$$

where $\ell(\cdot)$ denotes the specified (surrogate) loss function [4], such as *hinge* loss and *crossentropy* loss. Practically, the real distribution over (X, Z) is unknown to us, and what we can get is a set of training data $\{x_1, \cdots, x_N\}$ with the corresponding labels $\{z_1, \cdots, z_N\}$. Therefore, we often use the empirical risk $\hat{R}(f)$ to approximate the expected risk $R(f)$, namely

$$\hat{R}(f) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), z_i). \tag{5}$$

Note that, in label correction stage, we only get access to the aggregated labels, which may still suffer from label noise, so the empirical risk $\hat{R}(f)$ under clean labels is not computable. Fortunately, the following theorem provides us a feasible way to unbiasedly estimate the original risk $\hat{R}(f)$, which is

Theorem 1. *[33] Given the transactions $\{x_1, \cdots, x_N\}$ with the corrupted aggregated labels $\{y_1, \cdots, y_N\}$, one can approximate the original risk $\hat{R}(f)$ with true labels $\{z_1, \cdots, z_N\}$ by the following*

reweighted risk, namely

$$\hat{R}(f) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \beta(x_i, y_i)\ell(f(x_i), y_i), \tag{6}$$

where $\beta(x, y)$ is defined as $\frac{\hat{P}(X=x,Z=y)}{\hat{P}(X=x,Y=y)}$, $\hat{P}(\cdot)$ denotes the (estimated) probability, and $\ell(\cdot)$ is the original loss function.

Accordingly, given only the aggregated labels, we can still train a robust classifier by minimizing the empirical risk *w.r.t.* the weighted loss $\beta(x, y)\ell(f(x), y)$, which is

$$\hat{R}_{\mathrm{w}}(f) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \beta(x_i, y_i)\ell(f(x_i), y_i). \tag{7}$$

Intuitively, $\beta(x, y)$ assigns a large value to $\ell(\cdot)$ if the corresponding transaction x is correctly labeled by the aggregated label y, and a small value otherwise. Next, we show how to calculate the value of $\beta(x, y)$ for every example.

3.3.2 Estimating $\beta(x, y)$. Traditionally, $\beta(x, y)$ is interpreted as the density ratio that can be estimated by existing learning techniques [28, 29, 33]. These methods typically assume that the observed noisy label y of an example is irrelevant to its feature representation x. However, this is often not consistent with the practical cases as whether an example is correctly labeled should depend on its feature x. For our problem, the ambiguous transactions that are near the potential correct decision boundary are more likely to be mistakenly labeled. Therefore, in this paper, we relate $\beta(x, y)$ to both x and y to achieve a more realistic estimation. Fortunately, it is possible to get a small proportion of transactions $\{x'_1, \cdots, x'_M\}$ with the true labels $\{z'_1, \cdots, z'_M\}$ ($M \ll N$) that are provided by the human expert. Although this extra set of verified data may slightly increase the labeling costs (*e.g.,* payments for the human experts and time consumption), the quality of the estimated $\beta(x, y)$ can be critically enhanced, even when the aggregated labels still contain severe errors. Here, to facilitate the subsequent derivations, we first provide the following decomposition for $\beta(x, y)$, namely

$$\beta(x, y) = \frac{\hat{P}(X = x, Z = y)}{\hat{P}(X = x, Y = y)} = \frac{\hat{P}(X = x|Z = y)\hat{P}(Z = y)}{\hat{P}(X = x|Y = y)\hat{P}(Y = y)}, \tag{8}$$

which decomposes $\beta(x, y)$ into two parts, *i.e.,* $\frac{\hat{P}(X=x|Z=y)}{\hat{P}(X=x|Y=y)}$ and $\frac{\hat{P}(Z=y)}{\hat{P}(Y=y)}$. Now, we describe how to estimate their values.

For the first term $\frac{\hat{P}(X=x|Z=y)}{\hat{P}(X=x|Y=y)}$, it evaluates the tendency for the appearance of a transaction given its aggregated label y. For example, if a transaction x is considered to be genuine according to the aggregated label (*i.e.,* $Y = -1$), we have that $\hat{P}(X = x|Y = -1)$ will be large. If x is actually not a genuine transaction, the value of $\hat{P}(X = x|Z = -1)$ will be small, therefore the value of $\frac{\hat{P}(X=x|Z=-1)}{\hat{P}(X=x|Y=-1)}$ would be extremely small and the transaction x with a corrupted label will be less emphasized during training.

To compute the value of $\frac{\hat{P}(X=x|Z=y)}{\hat{P}(X=x|Y=y)}$, we make a general and appropriate assumption that the transactions with similar features should have similar ground-truth labels. In our implementation, the traditional clustering algorithm, such as k-means++, is deployed in measuring the similarity between different transactions. In this situation, transactions that belong to the same cluster are deemed to be similar. Mathematically, we have the following approximation:

$$\frac{\hat{P}(X = x|Z = y)}{\hat{P}(X = x|Y = y)} \approx \frac{\hat{P}(X \in \mathbb{C}_x|Z = y)}{\hat{P}(X \in \mathbb{C}_x|Y = y)}, \tag{9}$$

where $\mathbb{C}_x$ denotes the *cluster* (given by the clustering algorithm) that the transaction x belongs to. To be specific, we first deploy a clustering operation on all observed genuine transactions (*i.e.,* $\{(x, y)|y = -1\} \cup \{(x', z')|z' = -1\}$) and all observed fraudulent transactions (*i.e.,* $\{(x, y)|\ y = +1\} \cup \{(x', z')|z' = +1\}$), separately[4]. Then, with the help of the similarity assumption and the small verified clean set, the calculation of Eq. (9) is straightforward. For example, to estimate the probability $\hat{P}(x \in \mathbb{C}_x|Y = y)$, we calculate the proportion of corrupted transactions that belong to $\mathbb{C}_x$ in the original corrupted set, *i.e.,* $\hat{P}(x \in \mathbb{C}_x|Y = y) = \sum_{i=1}^{N}(\mathbb{1}[x_i \in \mathbb{C}_x]\mathbb{1}[y_i = y])/N$. Similarly, $\hat{P}(x \in \mathbb{C}_x|Z = y)$ can be estimated by calculating the proportion of the clean transactions belonging to $\mathbb{C}_x$ in the small clean set, *i.e.,* $\hat{P}(x \in \mathbb{C}_x|Z = y) = \sum_{i=1}^{M}(\mathbb{1}[x'_i \in \mathbb{C}_x]\mathbb{1}[z'_i = y])/M$. To sum up, we have

$$\frac{\hat{P}(X = x|Z = y)}{\hat{P}(X = x|Y = y)} \approx \frac{\sum_{i=1}^{M}(\mathbb{1}[x'_i \in \mathbb{C}_x]\mathbb{1}[z'_i = y])/M}{\sum_{i=1}^{N}(\mathbb{1}[x_i \in \mathbb{C}_x]\mathbb{1}[y_i = y])/N}. \tag{10}$$

Note that in the above process, the clustering operation actually involves the feature representation x, making the estimation of $\beta(x, y)$ in our method related to x.

For the second term $\frac{\hat{P}(Z=y)}{\hat{P}(Y=y)}$, it plays a role of distribution matching between Z and Y, which can be estimated by

$$\frac{\hat{P}(Z = y)}{\hat{P}(Y = y)} = \frac{\sum_{i=1}^{M}\mathbb{1}[z'_i = y]/M}{\sum_{i=1}^{N}\mathbb{1}[y_i = y]/N}, \tag{11}$$

where the numerator and denominator are considered with the proportion of the transactions labeled as y in the small clean set and that in the original corrupted set, respectively.

By combing Eq. (10) and Eq. (11), we can easily get the weight $\beta(x, y)$ for each transaction, and then we can train a robust classifier $g(x)$ by minimizing Eq. (7). Finally, we obtain the output of label correction stage, namely, the corrected labels $\{\hat{z}_1, \cdots, \hat{z}_N\}$, which are generated via $\hat{z}_i = g(x_i)$.

3.4 The Overall Algorithm

Algorithm 1 summarizes the overall algorithm that consists of two stages. In label aggregation stage, the paired weights to depict annotator quality $v_{k,-1}$ and $v_{k,+1}$ are both initialized to 1 at the beginning, such that the weighted voting in Step 9 degenerates to traditional majority voting. Otherwise, if the predicted labels from the robust classifier are available, the paired weights are updated in Step 7. Accordingly, we get access to the aggregated labels in Step 9. In label correction stage, k-means++ is respectively deployed on the data with genuine and fraudulent labels in Step 11, and the clustering results are used to calculate the weights in Step 12. In Step 13, the robust classifier is trained by minimizing Eq. (7), of which the results can be used to generate the corrected labels $\{\hat{z}_1, \cdots, \hat{z}_N\}$. These two stages execute iteratively for a pre-defined

[4]Note that both $\hat{P}(X = x|Z = y)$ and $\hat{P}(X = x|Y = y)$ depend on y, which indicates that the transactions with different labels $(+1/-1)$ should be clustered separately.

Algorithm 1 The overall algorithm.

Input: The historical transactions $\{x_i\}_{i=1}^N$ with the multi-sourced noisy annotations $\{y_i^1, \cdots, y_i^K\}_{i=1}^N$ and the verified transactions $\{x_i'\}_{i=1}^M$ with the true labels $\{z_i'\}_{i=1}^M$.

for $s \leftarrow 1$ to num_iters **do**
  //*LABEL AGGREGATION STAGE*
  if $s = 1$ **then**
    Initialize $v_{k,+1} = v_{k,-1} = 1$ for $k = 1, \cdots, K$;
  else
    Get the corrected labels $\{\hat{z}_i\}_{i=1}^N$ via $\hat{z}_i = g(x_i)$;
    Compute the weights $v_{k,+1}, v_{k,-1}$ for $k = 1, \cdots, K$ via Eq. (3);
  end if
  Get the aggregated labels $\{y_1, \cdots, y_N\}$ via Eq. (2);
  //*LABEL CORRECTION STAGE*
  Conduct k-means++ clustering on the data with observed label $y = z' = +1$ and $y = z' = -1$, separately;
  Calculate the weights $\beta(x, y)$ via Eq. (8);
  Train a robust classifier $g(x)$ by minimizing Eq. (7);
end for

Output: The robust classifier $g(x)$ for fraud detection.

iteration num_iters, and the final robust classifier $g(x)$ is adopted as the resulting fraud detector.

4 EXPERIMENTS

In this section, we examine the fraud detection ability of the proposed LAC on the real Alipay datasets by comparing our method with several existing representative approaches.

4.1 Datasets

In order to investigate the real-world fraud behaviors, we collect more than one million real-world online transaction records from two different types fraud scenarios of Alipay, which are respectively termed as *CCT* and *PAF* as mentioned in the Introduction.

In both datasets, each transaction is represented by a feature vector encoding the information such as basic attributes of accounts, the historical trading behaviors, *etc.* The multi-sourced noisy annotations are automatically provided according to different judgment criteria regarding a transaction. Note that, apart from the multiple noisy labels, it is possible to collect the true labels for a small proportion of transactions (*cf.*, Figure 1), within the acceptable budget and time. The basic characteristics of the two collected datasets, including the number of annotators, the dimension of the transaction vectors, and the size of the collected data, are provided in Table 1. As aforementioned, the label quality of the automatic annotation techniques is usually not satisfactory. To be specific, the average accuracy of positive (fraudulent) and negative (genuine) transactions over all annotators of *CCT* are 0.32 and 0.88, respectively. The average accuracy of positive (fraudulent) and negative (genuine) transactions over all annotators of *PAF* are 0.30 and 0.65, respectively. More details about the annotation quality can be found in Appendix B.1. We see that the label quality from the multiple sources is quite low, which poses great difficulty for a learning algorithm to fulfill accurate classification.

We conduct extensive experiments on these two fraud detection datasets in Alipay cases. Each dataset is randomly partitioned into 8:1:1 for training, validation, and test. Such partition is conducted ten times, and we report the average AUC (area under ROC curve) [34] of every compared method over ten independent trials. For the training set in each partition, the automatically generated noisy labels from multiple sources are used for model training. Moreover, we also randomly select a small proportion of transaction examples (0.1% by default) from the training set, and assign them with accurate labels to constitute the clean set. Note that the amounts of clean data are quite small in these two datasets, which are 560 out of totally 565,698 records on *CCT* and 590 out of totally 593,266 records on *PAF*, respectively.

Table 1: The characteristics of the two collected datasets from Alipay.

Dataset	#Annotator	#Dimension	#Transaction (#Neg. / #Pos.)
CCT	34	63	565,698 (548,450 / 17,248)
PAF	27	156	593,266 (501,834 / 91,432)

4.2 Model Instantiation

In Section 3.3, we propose a general risk-consistent estimator related to the weighted loss $\beta(x, y)\ell(f(x), y)$, which can be deployed in various basic classifiers. In this work, we instantiate our method with the following two backbones: **Multi-Layer Perceptron (MLP)**: it is a fully connected feedforward artificial neural network [21], which consists of at least three layers with different numbers of nodes, namely, the input layer, the hidden layer, and the output layer. **Gradient Boosting Decision Tree (GBDT)**: it is a gradient boosting algorithm that utilizes decision stumps or regression trees as weak classifiers. More specifically, in this work, the eXtreme Gradient Boosting (XGBoost) [12] is employed as an efficient and scalable implementation of GBDT.

4.3 Experimental Results

To evaluate the performance of the proposed two-staged fraud detection method on the collected datasets, intensive experiments have been done in comparison with some related methods that can also handle multi-sourced noisy annotations.

First of all, we describe the implementation details for our algorithm on each dataset. For both datasets, the number of clusters N_c in the k-means++ operation is set to 30 for examples with observed label $y = z' = +1$ and $y = z' = -1$, separately. For the parameters in MLP, we realize a 3-layer MLP with 128-dimension hidden layers and a *tanh* activation function. Then, we use mini-batch gradient ascent with a momentum of 0.9, a weight decay of 10^{-3}, a batch size of 256, and a learning rate of 0.1 for *CCT* and 0.01 for *PAF*. For the parameters in GBDT, we take the max tree depth as 6 for *CCT* and 9 for *PAF*, and other parameters such as η and λ in GBDT are set to the default values [45].

To demonstrate the effectiveness of our method, we compare it with various crowdsourcing algorithms on the two aforementioned datasets, which are

- Majority voting: a naïve method that takes the results of majority voting as the true labels for all examples.
- MeTal [41]: an ensemble method to produce reliable labels with weak supervision from diverse, multi-task sources having different granularities, accuracies, and correlations.

Table 2: Experimental results of the compared methods on *CCT* and *PAF* datasets. The best and second best results on each dataset are indicated in red and blue, respectively.

Dataset	*CCT*	*PAF*
Majority voting	0.780 ± 0.031	0.402 ± 0.023
MeTal [41]	0.810 ± 0.021	0.419 ± 0.023
CVL [31]	0.703 ± 0.035	0.457 ± 0.029
Yan2014 [55]	0.641 ± 0.023	0.409 ± 0.023
LAC_{MLP}	0.821 ± 0.014	0.497 ± 0.020
LAC_{GBDT}	0.829 ± 0.011	0.574 ± 0.031

- CVL [31]: a couple view learning approach for learning with multiple noisy labels, which alternately learns a data classifier and a label aggregator.
- Yan2014 [55]: a method that can estimate the true labels of examples and annotator expertise.
- LAC_{MLP} : the proposed method with MLP being the backbone in label correction stage.
- LAC_{GBDT}: the proposed method with GBDT being the backbone in label correction stage.

Table 2 summarizes the average AUC values of all compared methods on the test sets of the two datasets. We observe that our proposed approaches LAC_{MLP} and LAC_{GBDT} are consistently the two best methods among all compared methods for both datasets, with at least 0.019 and 0.117 AUC improvements over existing methods on *CCT* and *PAF*, respectively. Majority voting, MeTal [41], CVL [31], and Yan2014 [55] all perform unsatisfactorily due to the notoriously poor labeling quality of the annotators. By contrast, our method is able to handle the extremely noisy annotations with the help of a small set of verified data, and thus leading to superior results. It is worth noting that the performances of all compared methods on *PAF* are generally inferior to *CCT*, as the labeling quality of *PAF* is much worse as mentioned before. Note that, our method deploying the GBDT backbone performs better than that employing MLP, as GBDT has better generalization ability on imbalanced datasets than MLP as indicated in [18].

4.4 Verification of Label Correction

Note that the problem setting of label correction stage exactly coincides with the learning problem with noisy labels, so the proposed method for label correction stage can execute independently to deal with label noise problems. In this section, we investigate the effectiveness of the label noise processing method proposed in label correction stage by comparing it with typical algorithms that are robust to label noise. We follow the conventional setting of label noise learning, where the noisy labels provided by a single annotator are employed for training at each time. Similar to the experimental setting in Section 4.3, each dataset under a single annotator is randomly partitioned into 8:1:1 for training, validation, and test. A small proportion (0.1%) of training transactions are provided with true labels, which is used to assist training. The parameters of MLP and GBDT are kept consistent with the experimental setting in Section 4.3. The compared methods in dealing with label noise include

- BILN [13]: learning with bounded instance and label dependent label noise, which also deploys a few manually labeled clean data to assist training[5].
- RP [39]: rank pruning for robust classification with noisy labels.
- Re-weighting [33]: classification with noisy labels by importance re-weighting.
- ULE [38]: unbiased logistic estimator for learning with noisy labels.
- LR: traditional logistic regressor trained on noisy labels, which can be viewed as a baseline without tackling label noise.
- wMLP (ours): weighted multi-layer perceptron, which is the proposed label noise processing method in label correction stage with MLP being the backbone.
- wGBDT (ours): weighted gradient boosting decision tree, which refers to the proposed label noise processing method in label correction stage with GBDT being the backbone.

The average AUC values yielded by the compared algorithms on ten typical annotators of *CCT* dataset are shown in Table 3. Due to the page limit, the experimental results regarding verification of label correction on *PAF* dataset are deferred to Appendix B.3. We observe that our proposed approaches wMLP and wGBDT are superior to other compared methods in most cases. In particular, our methods perform much better than the method without noise correction, *i.e.*, LR, which demonstrates the significance of our methods in handling label noise. It is also shown that the canonical robust algorithms for dealing with label noise all fail here, as they cannot deal with the extremely low-quality labels as mentioned before. By contrast, the proposed method still performs well, verifying its effectiveness in dealing with extremely noisy labels.

4.5 Model Behavior Analyses

In this section, we investigate *i)* The AUC variation of our method *w.r.t.* the increase of the proportion of used verified transactions in label correction stage; *ii)* The parametric sensitivity of our approach to N_c, *i.e.*, the number of clusters constructed in label correction stage; *iii)* The convergence behavior of our method, since our model is trained via an iterative way (see Algorithm 1).

Firstly, we study the influence of the number of manually labeled clean data to the model output. As mentioned above, a small proportion of training transactions are provided with true labels in advance, which is crucial to improving the performance when the aggregated labels are still noisy. In this part, we provide a quantitative analysis on the influence of the proportion p_c of this small clean set to the entire training set. Figure 4 (a) and (b) show the experimental results of the proposed method on *CCT* and *PAF* with the proportion p_c changes within $\{0.01\%, 0.05\%, 0.1\%, 0.5\%, 1\%, 5\%\}$, respectively. We can see that, with the increase of the proportion, the AUC value of our method keeps rising and achieves a stable result when $p_c \geq 0.1\%$. It means that at most 0.1% clean examples are sufficient for our method to get a satisfactory performance. Such proportion is quite small, and manually labeling these transaction examples is acceptable in real-world fraud detection scenarios.

[5]In the experiments, we feed BILN with the same amount of clean data as that in our method to ensure a fair comparison.

Table 3: Experimental results of the compared label noise robust methods on ten typical annotators selected from *CCT* dataset. The best and second best results on each annotator are indicated in red and blue, respectively

Annotator	BILN [13]	RP [39]	Re-weighting [33]	ULE [38]	LR	wMLP(ours)	wGBDT(ours)
1	0.780 ± 0.046	0.648 ± 0.060	0.821 ± 0.031	0.614 ± 0.023	0.534 ± 0.020	0.832 ± 0.032	0.841 ± 0.033
2	0.719 ± 0.041	0.703 ± 0.023	0.735 ± 0.022	0.653 ± 0.032	0.510 ± 0.034	0.715 ± 0.021	0.770 ± 0.030
3	0.749 ± 0.026	0.724 ± 0.055	0.817 ± 0.036	0.697 ± 0.042	0.503 ± 0.045	0.805 ± 0.043	0.818 ± 0.045
4	0.735 ± 0.042	0.699 ± 0.028	0.780 ± 0.030	0.632 ± 0.029	0.558 ± 0.033	0.785 ± 0.032	0.805 ± 0.034
5	0.777 ± 0.040	0.697 ± 0.016	0.780 ± 0.015	0.624 ± 0.024	0.485 ± 0.015	0.781 ± 0.017	0.798 ± 0.018
6	0.635 ± 0.049	0.676 ± 0.004	0.620 ± 0.024	0.628 ± 0.023	0.438 ± 0.027	0.664 ± 0.025	0.633 ± 0.034
7	0.643 ± 0.049	0.711 ± 0.023	0.706 ± 0.028	0.662 ± 0.033	0.584 ± 0.023	0.711 ± 0.022	0.738 ± 0.024
8	0.691 ± 0.044	0.412 ± 0.034	0.487 ± 0.024	0.655 ± 0.035	0.602 ± 0.024	0.483 ± 0.023	0.695 ± 0.021
9	0.651 ± 0.018	0.448 ± 0.033	0.306 ± 0.032	0.636 ± 0.021	0.573 ± 0.023	0.571 ± 0.024	0.651 ± 0.033
10	0.702 ± 0.035	0.687 ± 0.026	0.782 ± 0.018	0.632 ± 0.034	0.483 ± 0.025	0.791 ± 0.027	0.798 ± 0.028
Average	0.708	0.640	0.683	0.643	0.540	0.713	0.753

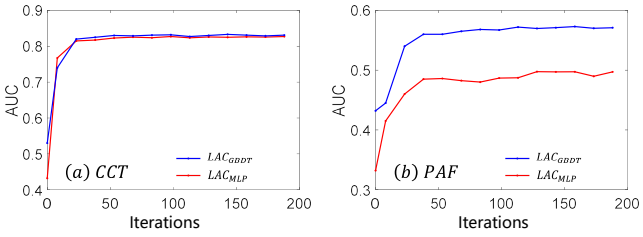

Figure 3: The convergence curves of our LAC on *CCT* and *PAF* datasets.

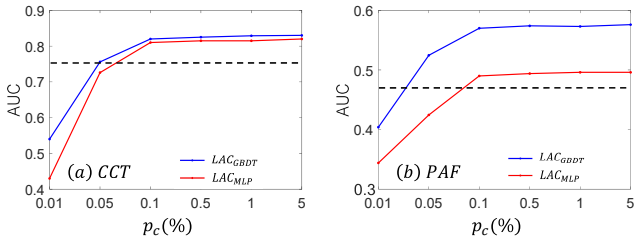

Figure 4: The influence of p_c to the performance of our LAC on *CCT* and *PAF* datasets, where the black dash line indicates the performance of the supervised model trained with only 0.1% clean data.

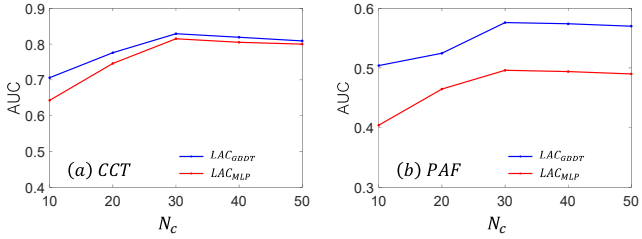

Figure 5: The parametric sensitivity of our LAC to N_c on *CCT* and *PAF* datasets.

Besides, the performance of the supervised backbone (*i.e.*, MLP) trained with only 0.1% clean data are also shown in Figure 4, *i.e.*, the black dash line, which performs far from satisfactory due to scarce supervision.

Secondly, it is worth noting that the number of clusters N_c in label correction stage needs to be tuned manually. In this part, we examine the parametric sensitivity of our approach to N_c. The experimental results on *CCT* and *PAF* with different N_c, *i.e.*, $N_c \in \{20, 30, 40, 50\}$, are reported in Figure 5 (a) and (b), respectively. From Figure 5, we observe that $N_c = 30$ is suggested for both *CCT* and *PAF* datasets to achieve the best performance.

Finally, it is also interesting to investigate the convergence behavior of our method. Figure 3 shows the convergence curves of our method during training, where our approach gradually converges to optimal and shows minor oscillation. We can have the conclusion that the iterative strategy is effective and label aggregation stage and label correction stage can benefit from each other.

5 CONCLUSION

In this paper, we propose a novel iterative two-staged fraud detection approach for protecting the capital safety of users, of which the target is to train a fraud detector under multi-sourced extremely noisy annotations. In label aggregation stage, we try to infer a reliable assignment for each transaction by modeling different annotators' labeling quality. In label correction stage, a label noise-robust algorithm is deployed to further correct the aggregated labels, with a handful of manually verified transactions. These two stages execute iteratively until convergence. Experimentally, we collected millions of transaction records in two different real-world fraud detection scenarios from Alipay, and the results on two collected datasets clearly demonstrate the effectiveness of the proposed method in detecting frauds. In the future, we plan to explore an active selection manner for choosing the very few yet critical transaction examples that need human annotation, and will also apply the proposed approach to more real-world fraud detection tasks.

ACKNOWLEDGEMENTS

This research is supported by NSF of China (Nos: 61973162 and 62006202), the Fundamental Research Funds for the Central Universities (Nos: 30920032202 and 30921013114), the RGC Early Career Scheme (No: 22200720), HKBU CSD Departmental Incentive Grant, and the Ant Financial Science Funds for Security Research of Ant Financial.

Table 4: Experimental results of the compared label noise robust methods on ten typical annotators selected from *PAF* dataset. The best and second best results on each annotator are indicated in red and blue, respectively.

Annotator	BILN [13]	RP [39]	Re-weighting [33]	ULE [38]	LR	wMLP(ours)	wGBDT(ours)
1	0.546 ± 0.019	0.613 ± 0.021	0.504 ± 0.023	0.561 ± 0.032	0.503 ± 0.032	0.615 ± 0.022	0.618 ± 0.024
2	0.502 ± 0.019	0.360 ± 0.019	0.381 ± 0.034	0.425 ± 0.023	0.421 ± 0.045	0.556 ± 0.022	0.600 ± 0.022
3	0.501 ± 0.020	0.392 ± 0.016	0.362 ± 0.034	0.434 ± 0.023	0.338 ± 0.020	0.505 ± 0.015	0.595 ± 0.023
4	0.424 ± 0.023	0.391 ± 0.022	0.434 ± 0.016	0.414 ± 0.024	0.330 ± 0.018	0.493 ± 0.019	0.460 ± 0.026
5	0.511 ± 0.043	0.468 ± 0.056	0.461 ± 0.011	0.445 ± 0.037	0.376 ± 0.032	0.387 ± 0.027	0.546 ± 0.034
6	0.489 ± 0.046	0.374 ± 0.027	0.457 ± 0.032	0.501 ± 0.020	0.358 ± 0.036	0.511 ± 0.033	0.533 ± 0.014
7	0.528 ± 0.045	0.438 ± 0.016	0.423 ± 0.024	0.630 ± 0.014	0.531 ± 0.016	0.610 ± 0.025	0.462 ± 0.021
8	0.519 ± 0.051	0.534 ± 0.033	0.532 ± 0.053	0.546 ± 0.058	0.424 ± 0.039	0.572 ± 0.046	0.614 ± 0.048
9	0.502 ± 0.011	0.402 ± 0.035	0.480 ± 0.029	0.530 ± 0.030	0.304 ± 0.029	0.615 ± 0.034	0.541 ± 0.033
10	0.540 ± 0.027	0.505 ± 0.025	0.540 ± 0.021	0.573 ± 0.030	0.347 ± 0.023	0.605 ± 0.023	0.613 ± 0.034
Average	0.458	0.447	0.453	0.506	0.393	0.547	0.558

A PROOF OF THEOREM 1

Proof. Here, we prove that the expected risk with true labels $R(f)$ can be unbiasedly estimated by the expected risk *w.r.t.* the weighted loss $\beta(x,y)\ell(f(x),y)$. To be specific, we have

$$\begin{aligned} R(f) &\overset{1}{=} \mathbb{E}_{(x,z)\sim(X,Z)}\left[\ell(f(x),z)\right] \\ &\overset{2}{=} \int_X\int_Z P(X=x,Z=z)\ell(f(x),z)dx\,dz \\ &\overset{3}{=} \int_X\int_Z P(X=x,Z=z)\frac{P(X=x,Y=z)}{P(X=x,Y=z)}\ell(f(x),z)dx\,dz \\ &\overset{4}{=} \int_X\int_Y P(X=x,Y=y)\frac{P(X=x,Z=y)}{P(X=x,Y=y)}\ell(f(x),y)dx\,dy \\ &\overset{5}{=} \mathbb{E}_{(x,y)\sim(X,Y)}\left[\frac{P(X=x,Z=y)}{P(X=x,Y=y)}\ell(f(x),y)\right] \\ &\overset{6}{=} \mathbb{E}_{(x,y)\sim(X,Y)}\left[\beta(x,y)\ell(f(x),y)\right], \end{aligned} \tag{12}$$

where the fourth equation holds with the variable substitution between y and z. Therefore, the empirical version of Eq. (12) is exactly Eq. (6) in the main manuscript, which concludes the proof. □

B EXPERIMENTAL DETAILS

In this section, we provide more experimental details that are not included in the main manuscript, namely, the evaluation metric, the labeling quality of ten typical annotators, and verification of label correction on *PAF*.

B.1 Labeling Quality of *CCT* and *PAF*

Table 5 shows the labeling quality of ten typical annotators selected from *CCT* and *PAF*, respectively. As we can see from Table 5, there exists extreme noise in the multi-sourced labels. It is also worth noting that labeling quality of the annotators in assigning "genuine" or "fraudulent" are quite different. For example, for annotator 1 in *PAF*, it effectively finds the majority of the fraudulent transactions, while the labeling quality on genuine transactions is extremely low. By contrast, for annotator 2, it identifies genuine transactions with high quality, but is confused about the fraudulent ones. Given the above observations, we see that the label quality from the multiple sources is quite low, which poses great difficulty for a learning algorithm to fulfill accurate classification.

Table 5: Labeling quality of some annotators from the two collected datasets in Alipay, where $Accu_{+1}$ and $Accu_{-1}$ respectively denote the class-wise accuracy in terms of the positive (fraudulent) and negative (genuine) transactions.

Dataset	*CCT*		*PAF*	
Annotator	$Accu_{+1}$	$Accu_{-1}$	$Accu_{+1}$	$Accu_{-1}$
1	0.260	0.988	0.954	0.056
2	0.359	0.956	0.030	0.954
3	0.461	0.914	0.992	0.001
4	0.262	0.698	0.502	0.369
5	0.427	0.904	0.600	0.322
6	0.509	0.731	0.225	0.694
7	0.748	0.633	0.436	0.415
8	0.264	0.821	0.297	0.667
9	0.788	0.190	0.139	0.925
10	0.971	0.022	0.489	0.661

B.2 Evaluation Metric

In this work, we use AUC as the evaluation metric because the amounts of positive examples and negative examples are quite imbalanced in both datasets as revealed by Table 1. Notably, it shows that a considerable amount of transactions are normal in general. For example, *CCT* consists of more than 565k genuine transactions, but only 17k transactions are fraudulent by contrast. As a result, AUC, which is one of the typical performance metrics for evaluating the class imbalanced learning algorithm, is adopted by our experiments, and the value of AUC can be interpreted as the probability that a classifier ranks fraudulent transactions higher than genuine ones [27]. In fact, such metric has also been widely employed in other existing fraud detection works [6, 11].

B.3 Verification of Label Correction on *PAF*

Table 4 summarizes the average AUC values of all compared label noise robust methods on *PAF*. Similar to *CCT*, we observe that the proposed approaches are superior to other compared methods in most cases, verifying the significance of the proposed method in handling label noise, especially on extreme label noise.

REFERENCES

[1] John Akhilomen. 2013. Data mining application for cyber credit-card fraud detection system. In *Industrial Conference on Data Mining*. Springer, 218–228.

[2] Shadi Albarqouni, Christoph Baur, Felix Achilles, Vasileios Belagiannis, Stefanie Demirci, and Nassir Navab. 2016. Aggnet: deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE T-MI* 35, 5 (2016), 1313–1321.

[3] John O Awoyemi, Adebayo O Adetunmbi, and Samuel A Oluwadare. 2017. Credit card fraud detection using machine learning techniques: A comparative analysis. In *ICCNI*. IEEE, 1–9.

[4] Shai Ben-David, David Loker, Nathan Srebro, and Karthik Sridharan. 2012. Minimizing the misclassification error rate using a surrogate convex loss. In *ICML*.

[5] Antonin Berthon, Bo Han, Gang Niu, Tongliang Liu, and Masashi Sugiyama. 2020. Confidence Scores Make Instance-dependent Label-noise Learning Possible. *ArXiv Preprint ArXiv:2001.03772* (2020).

[6] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland. 2011. Data mining for credit card fraud: A comparative study. *Decision Support Systems* 50, 3 (2011), 602–613.

[7] Christopher M Bishop. 2006. *Pattern recognition and machine learning*. springer.

[8] Bernardo Branco, Pedro Abreu, Ana Sofia Gomes, Mariana SC Almeida, João Tiago Ascensão, and Pedro Bizarro. 2020. Interleaved Sequence RNNs for Fraud Detection. In *SIGKDD*. 3101–3109.

[9] Fabrizio Carcillo, Yann-Ael Le Borgne, Olivier Caelen, and Gianluca Bontempi. 2017. An assessment of streaming active learning strategies for real-life credit card fraud detection. In *IEEE DSAA*. IEEE, 631–639.

[10] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, and Gianluca Bontempi. 2018. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization. *International Journal of Data Science and Analytics* 5, 4 (2018), 285–300.

[11] Nuno Carneiro, Goncalo Figueira, and Miguel Costa. 2017. A data mining based system for credit-card fraud detection in e-tail. *Decision Support Systems* 95 (2017), 91–101.

[12] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *SIGKDD*. 785–794.

[13] Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. 2020. Learning with bounded instance-and label-dependent label noise. In *ICML*.

[14] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28, 1 (1979), 20–28.

[15] Daniel de Roux, Boris Perez, Andrés Moreno, Maria del Pilar Villamil, and César Figueroa. 2018. Tax fraud detection for under-reporting declarations using an unsupervised machine learning approach. In *SIGKDD*. 215–222.

[16] Jun Du and Zhihua Cai. 2015. Modelling class noise with symmetric and asymmetric distributions. In *AAAI*.

[17] Benoît Frénay, Ata Kabán, et al. 2014. A comprehensive introduction to label noise. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

[18] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of Statistics* (2001), 1189–1232.

[19] Prasad Gabbur, Sharath Pankanti, Quanfu Fan, and Hoang Trinh. 2011. A pattern discovery approach to retail fraud detection. In *SIGKDD*. 307–315.

[20] Jyoti R Gaikwad, Amruta B Deshmane, Harshada V Somavanshi, Snehal V Patil, and Rinku A Badgujar. 2014. Credit Card Fraud Detection using Decision Tree Induction Algorithm. *IJITEE* 4, 6 (2014).

[21] Matt W Gardner and SR Dorling. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment* 32, 14-15 (1998), 2627–2636.

[22] Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *AAAI*. 1919–1925.

[23] Aritra Ghosh, Naresh Manwani, and PS Sastry. 2015. Making risk minimization tolerant to label noise. *Neurocomputing* 160 (2015), 93–107.

[24] Jacob Goldberger and Ehud Ben-Reuven. 2016. Training deep neural-networks using a noise adaptation layer. (2016).

[25] Chen Gong, Hong Shi, Tongliang Liu, Chuang Zhang, Jian Yang, and Dacheng Tao. 2019. Loss decomposition and centroid estimation for positive and unlabeled learning. *IEEE T-PAMI* 43, 3 (2019), 918–932.

[26] Chen Gong, Jian Yang, Jane J You, and Masashi Sugiyama. 2020. Centroid estimation with guaranteed efficiency: A general framework for weakly supervised learning. *IEEE T-PAMI* (2020).

[27] David J Hand. 2009. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *ML* 77, 1 (2009), 103–123.

[28] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. 2009. A least-squares approach to direct importance estimation. *JMLR* 10 (2009), 1391–1445.

[29] Takafumi Kanamori, Taiji Suzuki, and Masashi Sugiyama. 2010. Theoretical analysis of density ratio estimation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 93, 4 (2010), 787–798.

[30] Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. 2018. Learning from noisy singly-labeled data. In *ICLR*.

[31] Shikun Li, Shiming Ge, Yingying Hua, Chunhui Zhang, Hao Wen, Tengfei Liu, and Weiqiang Wang. 2020. Coupled-View Deep Classifier Learning from Multiple Noisy Annotators.. In *AAAI*. 4667–4674.

[32] Can Liu, Qiwei Zhong, Xiang Ao, Li Sun, Wangli Lin, Jinghua Feng, Qing He, and Jiayu Tang. 2020. Fraud Transactions Detection via Behavior Tree with Local Intention Calibration. In *SIGKDD*. 3035–3043.

[33] Tongliang Liu and Dacheng Tao. 2015. Classification with noisy labels by importance reweighting. *IEEE T-PAMI* 38, 3 (2015), 447–461.

[34] Jorge M Lobo, Alberto Jiménez-Valverde, and Raimundo Real. 2008. AUC: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography* 17, 2 (2008), 145–151.

[35] Yijing Luo, Bo Han, and Chen Gong. 2020. A Bi-level Formulation for Label Noise Learning with Spectral Cluster Discovery.. In *IJCAI*. 2605–2611.

[36] Aditya Krishna Menon, Brendan Van Rooyen, and Nagarajan Natarajan. 2016. Learning from binary labels with instance-dependent corruption. *ArXiv Preprint ArXiv:1605.00751* (2016).

[37] Kaixiang Mo, Erheng Zhong, and Qiang Yang. 2013. Cross-task crowdsourcing. In *SIGKDD*. 677–685.

[38] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *NeurIPS*. 1196–1204.

[39] Curtis G Northcutt, Tailin Wu, and Isaac L Chuang. 2017. Learning with confident examples: Rank pruning for robust classification with noisy labels. In *AAAI*.

[40] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*. 1944–1952.

[41] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training complex models with multi-task weak supervision. In *AAAI*, Vol. 33. 4763–4771.

[42] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *JMLR* 11, 4 (2010).

[43] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2014. Gaussian process classification and active learning with multiple annotators. In *ICML*. 433–441.

[44] Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *SIGKDD*. 614–622.

[45] https://github.com/dmlc/xgboost. 2020. XGBoost.

[46] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*. 839–847.

[47] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A Semi-supervised Graph Attentive Network for Financial Fraud Detection. In *ICDM*. IEEE, 598–607.

[48] Haibo Wang, Chuan Zhou, Jia Wu, Weizhen Dang, Xingquan Zhu, and Jilong Wang. 2018. Deep structure learning for fraud detection. In *ICDM*. IEEE, 567–576.

[49] Qizhou Wang, Bo Han, Tongliang Liu, Gang Niu, Jian Yang, and Chen Gong. 2021. Tackling Instance-Dependent Label Noise via a Universal Probabilistic Model. In *AAAI*. 10183–10191.

[50] Qizhou Wang, Jiangchao Yao, Chen Gong, Tongliang Liu, Mingming Gong, Hongxia Yang, and Bo Han. 2021. Learning with Group Noise. In *AAAI*. 10192–10200.

[51] Yang Wei, Chen Gong, Shuo Chen, Tongliang Liu, Jian Yang, and Dacheng Tao. 2019. Harnessing side information for classification under label noise. *IEEE T-NNLS* 31, 9 (2019), 3178–3192.

[52] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. 2010. The multidimensional wisdom of crowds. In *NeurIPS*. 2424–2432.

[53] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NeurIPS*. 2035–2043.

[54] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. 2019. Are Anchor Points Really Indispensable in Label-Noise Learning?. In *NeurIPS*. 6838–6849.

[55] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. 2014. Learning from multiple annotators with varying expertise. *ML* 95, 3 (2014), 291–327.

[56] Chuang Zhang, Chen Gong, Tengfei Liu, Xun Lu, Weiqiang Wang, and Jian Yang. 2020. Online Positive and Unlabeled Learning.. In *IJCAI*. 2248–2254.

[57] Chuang Zhang, Dexin Ren, Tongliang Liu, Jian Yang, and Chen Gong. 2019. Positive and Unlabeled Learning with Label Disambiguation.. In *IJCAI*. 4250–4256.

[58] Jing Zhang and Xindong Wu. 2018. Multi-label inference for crowdsourcing. In *SIGKDD*. 2738–2747.