Decouple then Combine: A Simple and Effective Framework for Fraud Transaction Detection

Pengwei Tang[†] TANGPWEI@163.COM
Renmin University of China

Huayi Tang[†] TANGH4681@GMAIL.COM
Renmin University of China

Wenhan Wang EZEWANG@TENCENT.COM
Tencent Inc.

Hanjing Su HANJINGSU@GMAIL.COM
Tencent Inc.

Yong Liu[*] LIUYONGGSAI@RUC.EDU.CN
Renmin University of China

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

With the popularity of electronic mobile and online payment, the demand for detecting financial fraudulent transactions is increasing. Although numerous efforts are devoted to tackling this problem, there are still two key challenges that are not well resolved, *i.e.*, the class imbalance ratio of test samples are extremely larger than that of training samples and amount of detected fraudulent transactions do not be considered. In this paper, we propose a simple and effective framework composed of majority and minority branches to address the above issues. The input samples of majority and minority branches come from vanilla and re-adjusted distribution, respectively. Parameters of each branch are optimized individually, by which the representation learning for majority and minority samples are decoupled. Besides, an extra loss re-weighted by amount is added in the majority branch to improve the recall amount of detected fraudulent transactions. Theoretical results show that under the proposed framework, minimizing the empirical risk is guaranteed to achieve small generalization risk on more imbalanced data with high probability. Experiments on real-world datasets from Tencent Wechat payments demonstrate that our framework achieves superior performance than competitive methods in terms of both number and money of detected fraudulent transactions.

Keywords: Imbalance Learning, Fraud Detection, Tabular Data

1. Introduction

With rapid development of digital economy, mobile payments have been integrated into people's daily lives. Accompanied by the widespread popularity of mobile payments, the occurrences of fraudulent transactions also increase significantly, resulting in large quantities amounts loss of users (Lin et al., 2021; Liu et al., 2021b,a). Therefore, financial fraud

. [†]Equal contribution. [*]Corresponding author.

detection has drawn increasing attention in recent years. There are two main tasks in financial fraud detection, *i.e.*, user-oriented task and transaction-oriented task. The former aims to judge whether a user is fraudulent or benign. The latter aims to distinguish fraudulent transactions from numerous transactions. In this work, we focus on the latter one. Both user-oriented and transaction-oriented financial detection tasks can be abstracted as imbalance classification (He and Garcia, 2009), one of the most fundamental but challenging problems in machine learning and data mining. Existing methods for imbalance classification generally fall into re-sampling approaches (Batista et al., 2004; Peng et al., 2019; Liu et al., 2020a), re-weighting approaches (Ren et al., 2018; Shu et al., 2019; Hu et al., 2019), ensemble learning approaches (Wang and Yao, 2009; Galar et al., 2013; Liu et al., 2020b), cost-sensitive learning approaches (Karakoulas and Shawe-Taylor, 1998; Fan et al., 1999), AUC optimization approaches (Qi et al., 2021; Yang et al., 2021), to name a few.

Although the above studies have made great progress in tackling the class imbalance problem, it is not trivial to apply them on real financial fraudulent detection scenarios. We emphasize the following three additional challenges. First, the number of fraudulent transactions is *extremely* smaller than that of benign transactions in real scenarios. Concretely, there is only *one* fraudulent transaction in millions of transactions. However, the ratio of majority to minority is commonly in the order of hundreds in previous studies. The extreme imbalance ratio seriously affects their effectiveness. Second, previous works (Cui et al., 2019; Cao et al., 2019; Shu et al., 2019) assume that class imbalance lies only in the training data. However, in real financial detection tasks, both training and test samples are class-imbalanced, and the test set has an even higher class-imbalance ratio than training data, due to the number of frauds will decrease under interception from online anti-fraud system. Third, the amounts vary by transaction, and it is required that the amounts of detected fraud transactions are as large as possible. Unfortunately, previous works only consider the number of recalled fraudulent transactions, leading to the model failing to discover some fraudulent transaction with large amounts. Besides, the data of payment transactions is commonly presented in tabular form, which is much more challenging than both image (Lin et al., 2017; Cui et al., 2019; Zhou et al., 2020) and graph-structured data (Lin et al., 2021; Liu et al., 2021b,a).

To overcome these challenges, in this paper, we propose a simple but effective method for detecting fraudulent transactions. We empirically found that re-weighting and re-balanced methods fail to handle the real-application data. Inspired by the conventional branch proposed in recent work (Zhou et al., 2020), our model is composed of majority-level and minority-level branches, which receive data from vanilla distribution and re-adjusted distribution, respectively. Since majority samples are dominant in vanilla distribution, the majority-level branch focuses on learning representations of majority samples. Differently, the minority samples are over-sampled in a re-adjusted distribution, by which the minority become dominant. It is worth mentioning that sharing weights in (Zhou et al., 2020) limit the representation ability of model, particularly on tabular data. To this end, the parameters of all branches are updated individually, so that the representations from majority and minority are decoupled. Besides, we additionally design loss re-weighted by amount for majority branch to improve the recall rate of amounts. After that, representations from different branches are aggregated via learnable weights to obtain the final prediction. Surprisingly, this simple model obtains superior performance on real-world data, which sheds

light on addressing this challenging issue for an industrial circle. Experiments are conducted on real-world datasets from Tencent Wechat payments, one of the largest electronic mobile and online payment platforms. The proposed framework significantly outperforms other competitive baselines. The main contributions of this work are summarized as follows:

- We propose to decouple the representation learning of majority and minority samples, which is simple to implement and empirically shown to be effective in addressing the extreme class imbalance problem.
- We provide a theoretical analysis on the proposed framework, which demonstrates that the proposed framework can achieve a small generalization risk on data with a higher class-imbalanced ratio.
- Experimental results on real-world data from Tencent Wechat payments verify the effectiveness of the proposed method.

2. Problem Formulation

Let $\mathcal{X}$ denote the feature space, $\mathcal{Y} = \{0, 1\}$ denote the space of class labels, and $\mathcal{M} \in \mathbb{R}_+ \cup \{0\}$ denote the space of transactions amounts. The training set $\widehat{\mathcal{D}} = \{(\boldsymbol{x}_i, y_i, m_i)\}_{i=1}^N$ is drawn from the distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y} \times \mathcal{M}$. For a transaction $(\boldsymbol{x}_i, y_i, m_i)$, the label $y_i = 0$ means it is benign while the label $y_i = 1$ means it is fraudulent, where m_i is the amount of this transaction. $m_i > 0$ when $y_i = 1$ and $m_i = 0$ when $y_i = 0$. Denote $\mathbb{I}(\cdot)$ as the indicator function. In our work, we consider the real-world fraudulent transaction detection, where the number of fraudulent payments is much less than the number of benign payments, *i.e.*,

$$\frac{\sum_{(\boldsymbol{x}_i, y_i, m_i) \in \widehat{\mathcal{D}}} \mathbb{I}(y_i = 0)}{\sum_{(\boldsymbol{x}_i, y_i, m_i) \in \widehat{\mathcal{D}}} \mathbb{I}(y_i = 1)} \gg 1. \tag{1}$$

Evaluation Metrics. Let $c(\boldsymbol{x}; \theta) \in [0, 1]$ denote a classifier parameterized by θ, which outputs a score measuring whether a payment is fraudulent. In real-world scenarios, to keep users from being defrauded, fraud detection can tolerate a certain number of benign payments being misclassified as fraudulent payments. Thus, the ratio of detected fraudulent payments to total fraudulent payments is commonly chosen as the evaluation metric. Due to the large scale of the payments, we can only focus on a small portion of the payments whose scores rank in the top. In our work, we use P_K and A_K as metrics, which are defined by:

$$P_K = \frac{\sum_{(\boldsymbol{x}_i, y_i) \in t(K)} y_i}{\sum_{(\boldsymbol{x}_i, y_i) \in \widehat{\mathcal{D}}_{\text{test}}} y_i}, \tag{2}$$

$$A_K = \frac{\sum_{(\boldsymbol{x}_i, y_i, m_i) \in t(K)} m_i}{\sum_{(\boldsymbol{x}_i, y_i, m_i) \in \widehat{\mathcal{D}}_{\text{test}}} m_i}, \tag{3}$$

where $\widehat{\mathcal{D}}_{\text{test}} = \{\boldsymbol{x}_i, y_i, m_i)\}_i^M$ denotes the test dataset. $t(K) \in \widehat{\mathcal{D}}_{\text{test}}$ denotes the subset of the test set whose scores rank in the top K:

$$t(K) = \left\{ (\boldsymbol{x}_i, y_i, m_i) | (\boldsymbol{x}_i, y_i, m_i) \in \widehat{\mathcal{D}}_{\text{test}}, c(\boldsymbol{x}_i; \theta) > \widehat{s}_K \right\}, \tag{4}$$

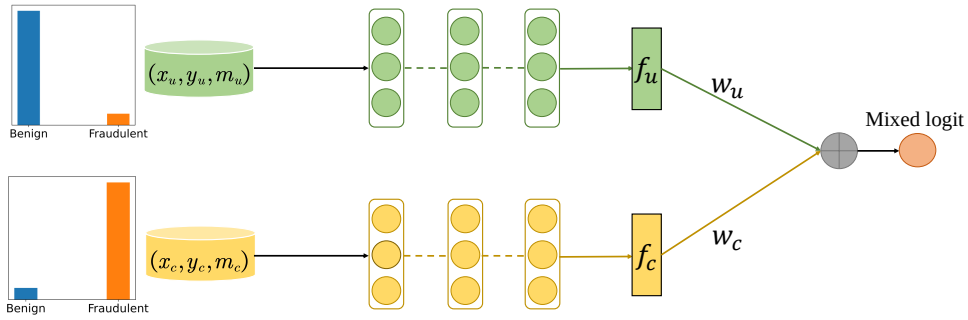

Figure 1: Illustration of the bi-collaborative learning framework. The green (orange) color denote the majority (minority) branch receiving samples from vanilla (re-adjusted) distribution.

where $\widehat{s}_K$ is exactly the K-th largest score in $\{c(\boldsymbol{x}_i;\theta)\}_i^M, \boldsymbol{x}_i \in \widehat{\mathcal{D}}_{\text{test}}$.

Note that in real-world scenarios, test set is still imbalanced, which is contrary to plenty of existing literature of imbalance learning (Cui et al., 2019; Cao et al., 2019; Shu et al., 2019). Thus, Equation (1) still holds in test set.

3. Our Proposed Method

In this section, we detail our proposed method, including data samplers, bi-collaborative learning framework, tri-collaborative learning framework and inference phase.

3.1. Data Samplers

In Figure 1, we visualize the bilateral branches framework for our bilateral collaborative learning method (BiCo). The two branches achieve different functions in imbalance learning via different data samplers. The majority branch adopts a instance-uniform data sampler for representation learning, while the minority branch adopts a class-reversed data samplers for re-balancing learning. The collaborative learning of these bilateral branches with different data samplers facilitates the learning in imbalanced scenarios.

First, we introduce the two data samplers in Figure 1, *i.e.*, instance-uniform data sampler and class-reversed data sampler. Recently, some literature show that in imbalance learning, using the cross entropy loss function over the original given imbalanced training dataset can lead to better semantic representation (Zhou et al., 2020; Yuan et al., 2021) as it can retain the original characteristics of the training data. The instance-uniform data samplers adopt the same distribution as the original training data. It is termed "instance-uniform data samplers" because it is equivalent to sampling each instance with same probability.

However, in imbalance training dataset, the "head" class dominates the loss function, making the classifier not pay enough attention to the "tail" class. Therefore, we introduce the class-reversed data sampler, which samples more "tail" class (fraudulent payments) than "head" class (benign payments). The class-reversed data sampler reverses the sampling

probability of the benign payments and fraudulent payments in training dataset. Let N_0 denote the number of benign payments and N_1 denote the number of fraudulent payments in the original training dataset. The class-reversed data sampler samples benign transactions with probability $\frac{N_1}{N_0+N_1}$ and samples fraudulent transactions with probability $\frac{N_0}{N_0+N_1}$.

3.2. Bi-Collaborative Learning Framework

Next, we elaborate the details of our bilateral branch framework for collaborative learning in Fraud transaction detection, which is termed BiCo. Both the upper branch and the lower branch are firstly fed to a multi-layer perceptron (MLP), which outputs two different feature representations. Let $(\boldsymbol{x}_u, y_u, m_u)$ and $(\boldsymbol{x}_c, y_c, m_c)$ denote the data generated by instance-uniform sampler and class-revversed sampler, respectively. Let $f_u(\boldsymbol{x}_u) \in \mathbb{R}^D$ denote the representation from instance-uniform data sampler (upper branch) and $f_c(\boldsymbol{x}_c) \in \mathbb{R}^D$ denote the representation from class-reversed data sampler (lower branch). These two MLPs adopt the same network architecture but do not share weights, which is one of key differences between BiCo and BBN (Zhou et al., 2020).

Furthermore, we propose a simple but effective approach to achieve collaborative learning of two branches. We introduce two mixed weights $\boldsymbol{w}_u \in \mathbb{R}^D$ and $\boldsymbol{w}_c \in \mathbb{R}^D$ to integrate $f_u(x_u)$ and $f_c(x_c)$. Then, we get the final mixed logit by a simple element addition, which is formulated as

$$z = \boldsymbol{w}_u^\top f_u(\boldsymbol{x}_u) + \boldsymbol{w}_c^\top f_c(\boldsymbol{x}_c), \tag{5}$$

where $z \in \mathbb{R}$ is the mixed logit. We adopt sigmoid function to calculate the predicted scores

$$s = \frac{1}{1+\exp(-z)}. \tag{6}$$

Then, the loss function of the proposed framework for collaborative learning is defined by

$$\mathcal{L} = l_{ce}(s, y_u) + l_{ce}(s, y_c), \tag{7}$$

where l_{ce} is the cross-entropy loss function:

$$l_{ce}(s, y) = -\left[ylog(s) + (1-y)log(1-s)\right]. \tag{8}$$

We adopt the mini-batch gradient descent to optimize the parameters of the two MLPs and the two mixed weights simultaneously.

In some real-world application scenarios, we are supposed to care more about the amount of the detected fraudulent payments (P_K) but not the number of the detected fraudulent payments, *i.e.*, A_K. A simple approach is to adopt the amounts as the coefficients for the cross-entropy loss function of the fraudulent payments, which is formulated as

$$l_{ce}^m = -\left[m \cdot ylog(s) + (1-y)log(1-s)\right]. \tag{9}$$

Thus, the loss function for increasing total amount of detected fraudulent payments is

$$\mathcal{L}^m = l_{ce}^{m_u}(s, y_u) + l_{ce}^{m_c}(s, y_c). \tag{10}$$

The bi-collaborative learning framework with Equation (10) is termed BiCo-A.

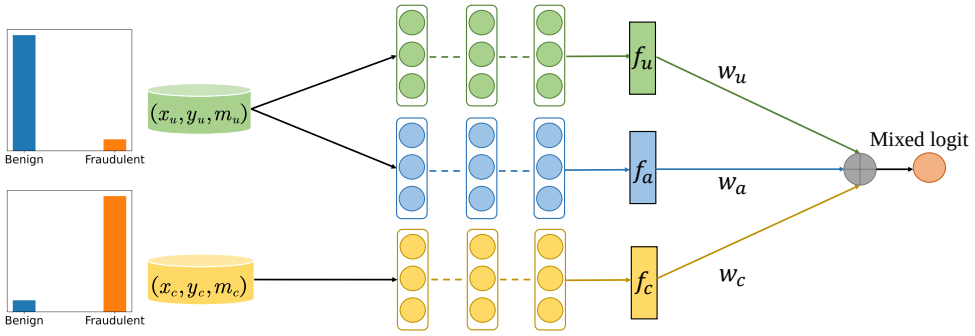

Figure 2: Illustration of the tri-collaborative learning framework. Compared with Figure 1, an extra branch denoted by blue is added to improve the amount of detected fraudulent payments.

3.3. Tri-collaborative Learning Framework

Although the use of the improved loss function $\mathcal{L}^m$ is effective for increasing A_k, it undermines the representation learning via original cross-entropy loss function, which hurts the capability of identifying detected fraudulent payments. To improve P_K and A_K simultaneously is a very challenging problem. To address this challenge, we propose the Tri-collaborative Learning Framework (TriCo) on top of BiCo, as shown in Figure 2.

In Figure 2, we adopt two data samplers, including instance-uniform data sampler and class-reversed data sampler, which is the same as the data samplers in BiCo. However, contrary to BiCo, the instance-uniform data sampler is adopted by two branches, *i.e.*, the upper branch and the middle branch. The upper branch of TriCo plays the same role as the upper branch of BiCo, which aims to learn good semantic representation via instance-uniform data sampler. The lower branch of TriCo is also the same the lower branch of BiCo. The middle branch is the difference between TriCo and BiCo. The mixed logit in the trilateral branch framework is defined as

$$z = \boldsymbol{w}_u^\top f_u(\boldsymbol{x}_u) + \boldsymbol{w}_a^\top f_a(\boldsymbol{x}_u) + \boldsymbol{w}_c^\top f_c(\boldsymbol{x}_c), \tag{11}$$

where $\boldsymbol{w}_u, \boldsymbol{w}_a, \boldsymbol{w}_c \in \mathbb{R}^D$ are the mixed weights of the upper branch, the middle branch and the lower branch, respectively.

On top of the overall loss function Equation (8) used in BiCo, we add a sub loss function, which is related to the middle branch. The middle branch aims to increase A_K. Thus, the middle branch adopt the improved cross-entropy loss function weighted by the amount of the fraudulent payments, i.e., Equation (9).

To summarize, the overall loss function adopted by TriCo is formulated as

$$\mathcal{L}^{Tri} = l_{ce}(s, y_u) + l_{ce}(s, y_c) + l_{ce}^{m_u}(s, y_u). \tag{12}$$

3.4. Inference Phase

For the bi-collaborative learning framework, during inference phase, the two branches are fed to the same input test data. It finally outputs a mixed logit with the information of

feature f_u and f_c. Thus, through original feature and re-balancing feature, the network can assign the fraudulent payments high scores.

For the tri-collaborative learning framework, during inference phase, the three branches are also fed to the same input test data. With original feature, re-balancing feature and the feature of increasing A_K, the network can also give the fraudulent payments high scores. Moreover, when only considering the fraudulent payments, the network can give fraudulent payments with more amount higher scores.

3.5. Theoretical Analysis

In this part, we provide theoretical analysis to the proposed framework. Without loss of generality, the analysis is oriented to BiCo whose loss function is given by Equation (7). Denote $\mathcal{D}_u$ and $\mathcal{D}_c$ the samples distribution of instance-uniform and class-reversed sampler. The training samples $(\boldsymbol{x}_u, y_u)$ and $(\boldsymbol{x}_c, y_c)$ are drawn independently from the joint distribution $\mathcal{D}_u \times \mathcal{D}_c$. Denote by $p = \frac{N_0}{N_1+N_0}$ the probability that $(\boldsymbol{x}_u, y_u)$ is benign in training sets, *i.e.*, $\Pr\{y_u = 1\} = p$. Let $\boldsymbol{x}_+ \sim \mathcal{D}_+$ and $\boldsymbol{x}_- \sim \mathcal{D}_-$ be the fraudulent (positive) and benign (negative) samples, respectively. $\mathcal{D}_+$ and $\mathcal{D}_-$ are the distribution of positive and negative samples.

Theorem 1 *Minimizing the loss defined in Equation (7) is guarantee to achieve a small generalization risk on more imbalanced data with high probability.*

Proof Denote by

$$\mathbb{E}_{(\boldsymbol{x}_u,y_u),(\boldsymbol{x}_c,y_c)\sim\mathcal{D}_u\times\mathcal{D}_c}\left[\mathcal{L}(\boldsymbol{x}_u, y_u, \boldsymbol{x}_c, y_c)\right]$$

the expectation of Equation (7). We have

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{x}_u,y_u),(\boldsymbol{x}_c,y_c)\sim\mathcal{D}_u\times\mathcal{D}_c}\left[\mathcal{L}(\boldsymbol{x}_u, y_u, \boldsymbol{x}_c, y_c)\right]\\
=&2p(1-p)\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{-z_u(\boldsymbol{x}_+)-z_c(\boldsymbol{x}_+)}\right)\right]\\
&+p^2\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{-z_u(\boldsymbol{x}_+)-z_c(\boldsymbol{x}_-)}\right)+\log\left(1+e^{z_u(\boldsymbol{x}_+)+z_c(\boldsymbol{x}_-)}\right)\right]\\
&+(1-p)^2\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{z_u(\boldsymbol{x}_-)+z_c(\boldsymbol{x}_+)}\right)+\log\left(1+e^{-z_u(\boldsymbol{x}_-)-z_c(\boldsymbol{x}_+)}\right)\right]\\
&+2p(1-p)\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{z_u(\boldsymbol{x}_-)+z_c(\boldsymbol{x}_-)}\right)\right]\\
\geq&2p(1-p)\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{-z_u(\boldsymbol{x}_+)-z_c(\boldsymbol{x}_+)}\right)\right]\\
&+2p(1-p)\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{z_u(\boldsymbol{x}_-)+z_c(\boldsymbol{x}_-)}\right)\right]\\
&+2\left(p^2+(1-p)^2\right)\log 2,
\end{aligned}$$

where $z_u(\boldsymbol{x}_+) = \boldsymbol{w}_u^\top f_u(\boldsymbol{x}_+)$, $z_u(\boldsymbol{x}_-) = \boldsymbol{w}_u^\top f_u(\boldsymbol{x}_-)$, $z_c(\boldsymbol{x}_+) = \boldsymbol{w}_c^\top f_c(\boldsymbol{x}_+)$, and $z_c(\boldsymbol{x}_-) = \boldsymbol{w}_c^\top f_c(\boldsymbol{x}_-)$. Denote by p' the probability that (x, y) is benign in test sets, *i.e.*, $\Pr\{y = 1\} = p'$. As we have mentioned, $p' < p$ holds in real-world financial fraudulent detection scenarios. Note that in the interference phase, both the majority and minority branch receive the same samples. Thus, the generalization risk is

$$\begin{aligned}
&\mathbb{E}_{(x,y)}\left[\mathcal{L}(\boldsymbol{x}, y, \boldsymbol{x}, y)\right]\\
=&p'\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{-z_u(\boldsymbol{x}_+)-z_c(\boldsymbol{x}_+)}\right)\right]\\
&+(1-p')\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{z_u(\boldsymbol{x}_-)+z_c(\boldsymbol{x}_-)}\right)\right].
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\mathbb{E}_{(x_u,y_u),(x_c,y_c)}\left[\mathcal{L}(\boldsymbol{x}_u,y_u,\boldsymbol{x}_c,y_c)\right]\\
>&2p'(1-p')\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{-z_u(\boldsymbol{x}_+)-z_c(\boldsymbol{x}_+)}\right)\right]+2\left(p^2+(1-p)^2\right)\log 2\\
&+2p'(1-p')\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{z_u(\boldsymbol{x}_-)+z_c(\boldsymbol{x}_-)}\right)\right]\\
\geq&2p'\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\Big[p'\log\left(1+e^{-z_u(\boldsymbol{x}_+)-z_c(\boldsymbol{x}_+)}\right)\\
&+(1-p')\log\left(1+e^{z_u(\boldsymbol{x}_-)+z_c(\boldsymbol{x}_-)}\right)\Big]+2\left(p^2+(1-p)^2\right)\log 2\\
>&2p'\mathbb{E}_{(x,y)}\left[\mathcal{L}(\boldsymbol{x},y,\boldsymbol{x},y)\right].
\end{aligned}$$

Thus, the expectation of $\mathcal{L}$ on test samples is the lower bound of the expectation of $\mathcal{L}$ on training samples, when neglecting the constant coefficient. According to the generalization analysis in (Mohri et al., 2018), the expectation term $\mathbb{E}_{(x_u,y_u),(x_c,y_c)}\left[\mathcal{L}(\boldsymbol{x}_u,y_u,\boldsymbol{x}_c,y_c)\right]$ can be upper bounded by the training loss $\mathcal{L}$ with high probability. This finishes the proof. ■

Theorem 1 shows that minimize $\mathcal{L}$ is beneficial to improving the generalization performance of model on testing samples when the test samples have a higher class-imbalance ratio than training samples. That is to say, if the model has a small training error, it can still achieve a small test error on more imbalanced samples with high probability. Now we briefly discuss the case that there is only the majority branch. It can be verified that

$$\begin{aligned}
&\mathbb{E}_{(x_u,y_u)}\left[\mathcal{L}(\boldsymbol{x}_u,y_u)\right]\\
=&p\mathbb{E}_{\boldsymbol{x}_+}\left[\log\left(1+e^{-z_u(\boldsymbol{x}_+)}\right)\right]+(1-p)\mathbb{E}_{\boldsymbol{x}_+,\boldsymbol{x}_-}\left[\log\left(1+e^{z_u(\boldsymbol{x}_-)}\right)\right],\\
&\mathbb{E}_{(x,y)}\left[\mathcal{L}(\boldsymbol{x},y)\right]\\
=&p'\mathbb{E}_{\boldsymbol{x}_+}\left[\log\left(1+e^{-z_u(\boldsymbol{x}_+)}\right)\right]+(1-p')\mathbb{E}_{\boldsymbol{x}_-}\left[\log\left(1+e^{z_u(\boldsymbol{x}_-)}\right)\right].
\end{aligned}$$

It can be found that minimizing $\mathbb{E}_{(x_u,y_u)}$ is not necessarily guarantee to minimize $\mathbb{E}_{(x,y)}\left[\mathcal{L}(\boldsymbol{x},y)\right]$, due to $p'<p$ and $1-p'>1-p$. Similar results can be obtained for the case that there is only the minority branch. Therefore, directly learning on the vanilla distribution or re-adjusted distribution do not guarantee to tackle the case that test sample are more imbalanced.

4. Experiments

4.1. Datasets and Metrics

Datasets. In experiments, we verify the effectiveness of our methods on a real-world transaction payment dataset from Tencent Wechat Group. Due to the need for confidentiality, the information about datasets is obscured. The training set is specially collected by Wechat Group, whose imbalance ratio is around a few tens. The training set was collected over a period of time. The test set contains 6 days of payment data, and we refer to them as Day1, Day2, Day3, Day4, Day5 and Day6. The test set is highly imbalanced, which is different from the setting of the majority of literature in computer vision (Zhou et al., 2020). The imbalance ratio of test set much exceeds that of training set and it may be up to millions. In our experiments, we will show the results of the six days and the average results.
Metrics. We adopt the metrics P_{20000}, A_{20000}, P_{50000} and A_{50000}. Please refer to Section 2 for the detailed definitions.

4.2. Baselines

In this section, we introduce some **strong** baseline method in computer vision to overcome data imbalance, which are detailed below.

Cross Entropy. Cross entropy loss is the most used loss function for classification tasks, which easily suffers from data imbalance.

Focal Loss (Lin et al., 2017). Focal loss adopts modulating weights to the cross-entropy loss in order to focus learning on hard samples.

Mixup (Zhang et al., 2018). Mixup is a data-agnostic data augmentation method, which convexly combines random pairs of training data and their associated targets.

CE-DRS and CE-DRW. These methods do not use bi-lateral branches but adopt two-stage training. In the first stage, it uses cross entropy loss for representation learning; in the second stage, it adopts re-balance learning to overcome data imbalance. Following the previous literature (Cao et al., 2019), we use re-sampling (CE-DRS) and re-weighting (CE-DRW) to achieve re-balance in the second stage.

LDAM and LDAM-DRW (Cao et al., 2019). The label-distribution-aware margin loss (LDAM) encourages larger margins for minority classes to address class imbalance. The LDAM-DRW is the combination of LDAM and re-weighting, which trains network first with vanilla LDAM loss and then transitions to using LDAM loss with a re-weighting schedule.

Class-balanced Loss (Cui et al., 2019). The class-balanced loss is to re-weight loss inversely with the effective number of samples per class. We adopt class-balanced cross entropy loss (CB-CE) and class-balanced focal loss (CB-Focal).

BBN (Zhou et al., 2020). BBN is the most relevant to our work. BBN also adopts a bilateral-branch network with instance-uniform and class-revered sampler for representation learning and classifier learning. The key difference between our method and BBN is that the two branches of BBN apply the same network but the two branches of our method adopt two difference networks with the same network architecture.

The Variant Loss of the Above Methods for Increasing A_K. In order to improve A_K, we adopt the amount m as the coefficient of the loss function of the fraudulent payments for the above method, which is similar to Equation (9). We use "-A" to represent these variants. including Cross Entropy-A, Focal Loss-A, Mixup-A, CE-DRS-A, CE-DRW-A, LDAM-A, LDAM-DRW-A, CB-CE-A, CB-Focal-A and BBN-A.

4.3. Implementation Details

The experiments is implemented by TensorFlow. For fair comparisons, the base settings are the same for all methods. The mini-batch size is set to 512. The training epochs is set to 20. The architecture of MLPs is set to 2048-512-128-1 except LDAM and BBN. LDAM Loss and its variant adopt the architecture 2048-512-128-2. BBN and its variant adopt the architecture 2048-512-128-64-1, where the architecture part 2048-512-128 is used for sharing weights. We adopt Adam optimizer to train the MLP. The initial learning rate is 0.0001.

4.4. The Effect of Our Method

In Tables 1 to 4, we show the performance comparisons between our methods and the baseline methods over Day1-Day4. Tables 1 and 3 show the performance comparisons between BiCo and the baseline methods without considering the amount of fraud payments.

Table 1: Performance comparisons between our methods and baselines with considering amount (Day1 - Day2). The best results are denoted by bold.

Date	Day 1				Day 2			
Method	P_{20000}	A_{20000}	P_{50000}	A_{50000}	P_{20000}	A_{20000}	P_{50000}	A_{50000}
Cross Entropy	21.8	10.9	30.8	18.2	41.9	36.7	49.6	45.7
Focal Loss	24.8	11.9	27.1	12.4	35.0	20.9	48.7	41.0
Mixup	18.0	9.2	28.6	23.1	40.2	31.6	51.3	**46.8**
CE-DRS	0.0	0.0	0.8	0.2	0.9	5.1	0.9	5.1
CE-DRW	22.6	11.8	30.8	18.7	36.8	30.4	47.0	42.5
LDAM	21.1	7.8	27.8	14.5	**41.9**	**41.8**	49.6	44.6
LDAM-DRW	20.3	10.5	25.6	12.3	38.5	25.7	44.4	31.3
CB-CE	18.8	15.0	26.3	**22.7**	30.8	17.7	43.6	37.5
CB-Focal	19.5	10.0	26.3	14.2	33.3	27.2	38.5	28.8
BBN	3.8	2.1	5.3	5.8	6.0	8.2	7.7	17.5
BiCo	**25.6**	**15.9**	**32.3**	20.2	29.1	27.1	**53.0**	45.4

Table 2: Performance comparisons between our methods and baselines considering amount (Day1 - Day2). The best results are denoted by bold.

Date	Day 1				Day 2			
Method	P_{20000}	A_{20000}	P_{50000}	A_{50000}	P_{20000}	A_{20000}	P_{50000}	A_{50000}
Cross Entropy-A	25.6	12.8	37.6	30.5	41.9	39.5	50.4	45.7
Focal Loss-A	26.3	12.3	32.3	20.5	43.6	36.1	53.0	47.0
Mixup-A	26.3	17.6	34.6	27.6	31.6	33.0	41.9	43.6
CE-DRS-A	4.5	1.9	8.3	4.2	0.0	0.0	0.0	0.0
CE-DRW-A	27.8	13.2	33.1	18.9	40.2	26.6	53.0	45.2
LDAM-A	21.8	14.4	33.8	24.0	40.2	38.2	51.3	46.4
LDAM-DRW-A	24.1	11.7	29.3	14.9	41.9	41.4	52.1	46.1
CB-CE-A	24.1	16.1	33.1	21.7	28.2	24.4	49.6	34.1
CB-Focal-A	22.6	16.6	30.8	19.4	30.8	22.2	42.7	32.6
BBN-A	6.0	13.7	9.8	21.0	2.6	8.9	6.0	14.5
BiCo	25.6	15.9	32.3	20.2	29.1	27.1	53.0	45.4
BiCo-A	**31.6**	**27.3**	**39.1**	**33.3**	30.8	39.0	47.9	**51.1**
TriCo	29.3	17.9	35.3	25.4	**47.9**	**44.1**	**58.1**	47.9

Tables 2 and 4 show the performance comparisons between our three methods and the baseline methods with considering the amount of fraud payments. In Table 5, the average experiment results are presented. **The experiment results of Day5-Day6 are detailed in the supplementary. In Tables 5 to 7, the experiment results are the average of 6 Days.**

In Tables 2 and 4, we can observe that BiCo, a method which also does not consider the money of fraudulent payments, obtains the best results in most cases. According to

Table 3: Performance comparisons between our methods and baselines with considering amount (Day3 - Day4). The best results are denoted by bold.

Date	Day 3				Day 4			
Method	P_{20000}	A_{20000}	P_{50000}	A_{50000}	P_{20000}	A_{20000}	P_{50000}	A_{50000}
Cross Entropy	17.9	9.7	25.9	19.2	12.2	6.5	21.6	11.6
Focal Loss	17.9	8.0	25.9	12.1	10.8	6.0	20.3	10.6
Mixup	11.6	5.5	26.8	16.3	10.8	4.2	29.7	15.5
CE-DRS	0.9	0.9	1.8	1.4	0.0	0.0	0.0	0.0
CE-DRW	18.8	8.9	27.7	14.4	12.2	7.3	25.7	13.8
LDAM	20.5	13.6	27.7	16.6	17.6	8.6	27.0	13.8
LDAM-DRW	17.0	9.3	31.2	18.7	**24.3**	13.0	27.0	14.6
CB-CE	21.4	**21.4**	32.1	**28.3**	18.9	10.1	21.6	11.3
CB-Focal	14.3	9.2	24.1	14.1	18.9	9.6	25.7	13.8
BBN	3.6	4.7	3.6	4.7	2.7	2.6	4.1	3.1
BiCo	**27.7**	13.7	**41.1**	17.7	**24.3**	**13.6**	**40.5**	**26.2**

Table 4: Performance comparisons between our methods and baselines without considering amount (Day3 - Day4). The best results are denoted by bold.

Date	Day 3				Day 4			
Method	P_{20000}	A_{20000}	P_{50000}	A_{50000}	P_{20000}	A_{20000}	P_{50000}	A_{50000}
Cross Entropy-A	15.2	9.8	22.3	13.7	14.9	9.6	18.9	12.3
Focal Loss-A	15.2	10.1	27.7	16.1	12.2	6.9	16.2	9.6
Mixup-A	17.9	16.7	23.2	19.4	12.2	7.6	17.6	10.7
CE-DRS-A	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CE-DRW-A	15.2	10.3	33.0	18.1	10.8	7.0	21.6	12.0
LDAM-A	14.3	7.0	25.0	15.0	8.1	4.5	25.7	13.6
LDAM-DRW-A	13.4	8.7	24.1	13.1	17.6	8.3	21.6	11.1
CB-CE-A	14.3	10.2	22.3	12.9	10.8	6.8	23.0	16.2
CB-Focal-A	9.8	5.0	17.9	8.9	9.5	6.4	14.9	9.1
BBN-A	0.0	0.0	10.7	22.4	1.4	0.8	5.4	7.2
BiCo	27.7	13.7	41.1	17.7	**24.3**	13.6	**40.5**	**26.2**
BiCo-A	25.0	**20.1**	**45.5**	**50.8**	14.9	**14.5**	20.3	18.1
TriCo	**30.4**	16.6	**45.5**	23.9	**24.3**	13.9	32.4	17.4

Table 5, BiCo outperforms the baseline methods without considering the money of fraudulent payments with a significant margin. We can also observe that these strong baselines in computer vision have very limited improvement or even worse performance compared to the performance of Cross Entropy. It means that compared to image data, the real-world payment data have many different characteristics including the extreme high imbalance of the test data and the tabular form in which the payments present, leading to ineffectiveness of the methods in computer vision. Note that the BBN method similar to BiCo also has

Table 5: Performance comparisons between ours methods and baseline methods (Average of Day1 - Day 6). The best results are denoted by bold.

	Average Results			
Method	P_{20000}	A_{20000}	P_{50000}	A_{50000}
Cross Entropy	21.8	16.8	32.3	26.5
Cross Entropy-A	22.6	19.8	32.7	32.0
Focal Loss	20.0	13.6	29.4	22.6
Focal Loss-A	22.1	16.9	31.7	25.2
Mixup	18.2	15.4	30.8	26.9
Mixup-A	20.7	18.2	30.7	30.7
CE-DRS	0.3	1.0	0.7	1.2
CE-DRS-A	0.8	0.3	1.6	0.8
CE-DRW	21.6	16.3	33.7	25.7
CE-DRW-A	22.7	18.6	35.3	27.9
LDAM	23.6	21.1	33.6	30.0
LDAM-A	19.4	16.0	31.7	24.9
LDAM-DRW	23.0	13.7	32.3	24.1
LDAM-DRW-A	22.1	17.8	29.7	22.7
CB-CE	21.9	20.8	31.0	28.7
CB-CE-A	19.1	16.8	32.5	28.2
CB-Focal	20.7	17.3	30.4	26.3
CB-Focal-A	16.7	13.9	26.3	21.6
BBN	4.2	4.9	5.9	9.5
BBN-A	4.2	10.2	9.6	19.9
BiCo	**27.7**	19.8	**43.9**	30.5
BiCo-A	27.1	**33.0**	39.7	**45.9**
TriCo	**32.3**	25.0	**45.5**	34.8

very poor performance, which is probably due to the operation of sharing weights. MLP does not have the same powerful performance in tabular data as performance of Convolutional neural network (CNN) in image data. We provide more discussions and experiments in our ablation study to further investigate the difference between our methods and BBN.

In Tables 2 and 4, we compare our three methods with the baseline methods while considering the amount of the fraudulent payments. We can observe that our methods have the best performance in most cases. In Table 5, both BiCo-A and TriCo outperform the baseline methods with considering the money of fraudulent payments in all metrics. Combined with Table 1 and Table 2 or Table 3 and Table 4, we can observe that considering the amount of the fraudulent payments, i.e., adopting the amount as the coefficient of the loss of the fraudulent payments like Equation (9), help increase A_{20000} and A_{50000}, which means that the networks have greater capability of identifying the fraudulent payments with high amount. In Table 5, we can observe that directly replacing the normal cross entropy loss with Equation (9), *i.e.*, BiCo-A, can increase A_{20000} and A_{50000}, but it undermines

Table 6: Performance comparisons between BiCo and BBN under different adaptor strategies (Average of Day1 - Day6).

α	Method	P_{20000}	A_{20000}	P_{50000}	A_{50000}
0.5	BiCo	27.7	19.8	43.9	30.5
	BBN	10.9	11.1	14.6	13.2
$Beta(0.2, 0.2)$	BiCo	30.1	21.9	43.5	32.4
	BBN	7.7	7.0	12.8	10.8
$(\frac{t}{T_{max}})^2$	BiCo	25.5	14.3	40.2	23.7
	BBN	4.1	4.4	5.1	4.8
$1 - \frac{t}{T_{max}}$	BiCo	28.5	21.8	41.6	33.6
	BBN	19.4	13.4	24.0	16.9
$cos(\frac{t}{T_{max}} \cdot \frac{\pi}{2})$	BiCo	30.1	21.1	39.4	26.3
	BBN	4.3	5.3	8.1	9.1
$1 - (\frac{t}{T_{max}})^2$	BiCo	25.9	18.8	40.2	26.8
	BBN	4.5	7.0	6.7	8.5

Table 7: Performance comparisons between TriCo and TriCo-CR (Average of Day1 - Day6).

Method	P_{20000}	A_{20000}	P_{50000}	A_{50000}
TriCo	32.3	25.0	45.5	34.8
TriCo-CR	28.5	27.9	42.9	41.7

P_{20000} and P_{50000}. TriCo, the method with trilateral branch framework, can help increase A_{20000} and A_{50000} and not undermine P_{20000} and P_{50000}.

5. Ablation study

5.1. More Study about BiCo and BBN

The BBN method (Zhou et al., 2020) seems similar to BiCo (ours). BiCo and BBN both use instance-uniform data sampler and class-reversed data sampler, and both employ the bi-lateral branch framework. In BiCo and BBN, the branch with instance-uniform data sampler both aims to achieve representation learning based on the original characteristic of the training data; the branch with class-revered data sampler both aim to overcome the high imbalance. For BBN, its loss function is formulated as

$$\mathcal{L} = \alpha l_{upper} + (1 - \alpha) l_{lower},$$

where α is a trade-off between l_{upper} and l_{lower}, and BBN adopts a parabolic decay strategies $\alpha = 1 - (\frac{t}{T_{max}})^2$ to adjust α. According to paper proposed BBN method (Zhou et al., 2020), it investigates different adaptor strategies to adjust α dynamically. Thus, we also investigate these strategies on BiCo and BBN. As shown in Table 6. We present the comparisons between Bico and BBN on six different adaptors strategies. We can observe that BiCo outperforms BBN under all six different strategies with significant margins. BBN has very

poor performance under all six adaptor strategies compared to Cross Entropy from Table 5. However, BiCo performs well under all conditions. Thus, the reason why BBN perform poorly but BiCo performs well is that MLP do not have the powerful capability to capture representation information and re-balanced information in one network in tabular data as CNN in image data. Thus, the operation of sharing weights in BBN makes it suffer from serious performance degradation.

5.2. More about Tri-collaborative Framework

The extra middle branch in Figure 2 adopts the cross-entropy considering the amount of the fraudulent payments and it applies the instance-uniform data sampler. In this section, we investigate the middle branch with class-revered data sampler, and we term such tricollaborative learning framework TriCo-CR. As shown in Table 7, we can observe that TriCo-CR also performs well. For P_{20000} and P_{50000}, TriCo performs better than TriCO-CR. In contrast, for A_{20000} and P_{50000}, TriCo-CR outperforms TriCo. The extra middle branch of TriCo considers instance-uniform data, which can help improve the representation ability, and thus improve the capability of identifying fraudulent payments. The extra middle branch of TriCo-CR considers class-reversed data sampler where fraudulent payments dominate the mini-batch, and mainly cares about the fraudulent payments with high amount. Thus, TriCo-CR has high value of A_{20000} and A_{50000}.

6. Conclusion

In this paper, we propose a simple and effective model to detect fraudulent transactions in electronic mobile and online payment platforms. With the key insight that decoupling the representation learning of majority and minority samples, our framework contains parallel branches with individual parameters to learn representations for majority and minority samples. Besides, we additionally add a loss re-weighted by amount to improve the recall amount of detected fraudulent transactions. Theoretical analysis shows that the proposed framework is applicable in the scenarios where test samples have a higher class-imbalanced ratio. Experiments on real-world datasets show that the proposed framework surpasses competitive methods. Our future work includes deploying the proposed framework at Wechat payment platform and exploring how to apply it on graph-structured data.

Acknowledgments

We thank the anonymous reviewers for their valuable and constructive suggestions and comments. This work is supported by the Tencent WeChat Rhino-Bird Focused Research Program; the Beijing Natural Science Foundation (No.4222029); the National Natural Science Foundation of China (NO.62076234); the National Key Research and Development Project (No.2022YFB2703102); the "Intelligent Social Governance Interdisciplinary Platform, Major Innovation Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China"; the Beijing Outstanding Young Scientist Program (NO.BJJWZYJH012019100020098); the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (NO.2021030199).

References

Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in Neural Information Processing Systems*, 32, 2019.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

Wei Fan, Salvatore J. Stolfo, Junxin Zhang, and Philip K. Chan. Adacost: Misclassification cost-sensitive boosting. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 97–105, 1999.

Mikel Galar, Alberto Fernández, and Edurne. Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition*, 46(12):3460–3471, 2013.

Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

Zhiting Hu, Bowen Tan, Russ R Salakhutdinov, Tom M Mitchell, and Eric P Xing. Learning data manipulation for augmentation and weighting. *Advances in Neural Information Processing Systems*, 32, 2019.

Grigoris Karakoulas and John Shawe-Taylor. Optimizing classifers for imbalanced training sets. *Advances in Neural Information Processing Systems*, 11, 1998.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

Wangli Lin, Li Sun, Qiwei Zhong, Can Liu, Jinghua Feng, Xiang Ao, and Hao Yang. Online credit payment fraud detection via structure-aware hierarchical recurrent neural network. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 3670–3676, 2021.

Can Liu, Li Sun, Xiang Ao, Jinghua Feng, Qing He, and Hao Yang. Intention-aware heterogeneous graph attention networks for fraud transactions detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3280–3288, 2021a.

Yang Liu, Xiang Ao, Qiwei Zhong, Jinghua Feng, Jiayu Tang, and Qing He. Alike and unlike: Resolving class imbalance problem in financial credit risk assessment. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, page 2125–2128, 2020a.

Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference*, page 3168–3177, 2021b.

Zhining Liu, Pengfei Wei, Jing Jiang, Wei Cao, Jiang Bian, and Yi Chang. Mesa: boost ensemble imbalanced learning with meta-sampler. *Advances in neural information processing systems*, 33:14463–14474, 2020b.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT press, 2018.

Minlong Peng, Qi Zhang, Xiaoyu Xing, Tao Gui, Xuanjing Huang, Yu-Gang Jiang, Keyu Ding, and Zhigang Chen. Trainable undersampling for class-imbalance learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4707–4714, 2019.

Qi Qi, Youzhi Luo, Zhao Xu, Shuiwang Ji, and Tianbao Yang. Stochastic optimization of areas under precision-recall curves with provable convergence. *Advances in neural information processing systems*, 34:1752–1765, 2021.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in Neural Information Processing Systems*, 32, 2019.

Shuo Wang and Xin Yao. Diversity analysis on imbalanced data sets by using ensemble models. In *2009 IEEE symposium on computational intelligence and data mining*, pages 324–331. IEEE, 2009.

Zhiyong Yang, Qianqian Xu, Shilong Bao, Yuan He, Xiaochun Cao, and Qingming Huang. When all we need is a piece of the pie: A generic framework for optimizing two-way partial auc. In *International Conference on Machine Learning*, pages 11820–11829. PMLR, 2021.

Zhuoning Yuan, Zhishuai Guo, Nitesh Chawla, and Tianbao Yang. Compositional training for end-to-end deep auc maximization. In *International Conference on Learning Representation*, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9716–9725, 2020.