Deep Representation Learning With Full Center Loss for Credit Card Fraud Detection

Zhenchuan Li, Guanjun Liu, *Senior Member, IEEE*, and Changjun Jiang

***Abstract*—Credit card fraud detection is an important study in the current era of mobile payment. Improving the performance of a fraud detection model and keeping its stability are very challenging because users' payment behaviors and criminals' fraud behaviors are often changing. In this article, we focus on obtaining deep feature representations of legal and fraud transactions from the aspect of the loss function of a deep neural network. Our purpose is to obtain better separability and discrimination of features so that it can improve the performance of our fraud detection model and keep its stability. We propose a new kind of loss function, full center loss (FCL), which considers both distances and angles among features and, thus, can comprehensively supervise the deep representation learning. We conduct lots of experiments on two big data sets of credit card transactions, one is private and another is public, to demonstrate the detection performance of our model by comparing FCL with other state-of-the-art loss functions. The results illustrate that FCL outperforms others. We also conduct experiments to show that FCL can ensure a more stable model than others.**

***Index Terms*—Credit card fraud detection, loss function, performance stability, representation learning.**

I. Introduction

FRAUDSTERS and detectors of credit card transactions keep a dynamic game process for a long time. Especially in the current Internet times, transaction fraud events take place more frequently than ever before and result in substantial economic losses. The Nilson report delivered comprehensive research about the situation of worldwide card fraud. The total financial losses from credit card fraud reached \$21.84 billion in 2015, increased to \$24.71 billion in 2016, and were over \$27 billion in 2017. What even worse is that the global card fraud losses will keep increasing year by year and possibly reach \$31.67 billion in 2020 [1].

Therefore, an effective fraud detection system is essential for banks and financial institutions to detect or monitor transactions online. Different fraud detection systems have the same target, which is to mine suspicious transaction patterns from a number of transaction logs so that they can be used to detect or monitor an incoming transaction. It has been demonstrated that machine learning is extremely effective for mining these patterns, which can be viewed as a task of supervised binary classification [2]–[7]. In other words, abundant transaction records can be used to train a well-performed classifier for identifying fraud transactions. Although machine learning has achieved remarkable success in detecting fraud transactions, the improvement of a fraud detection system will never stop, and even a little improvement can reduce huge financial losses.

Machine-learning-based credit card fraud detection is much more challenging than traditional binary classification tasks such as image classification. There are two main reasons: class imbalance in data sets and dynamic changes in behaviors of users and fraudsters [2], [8]. On the one hand, there is an extremely small number of fraud transaction records typically available over all transaction records, and thus, this affects the performance of a supervised classification approach seriously [9]. Fortunately, many efforts have been made to handle this problem, such as the sampling-based method [10] and cost-based method [11]. On the other hand, fraudsters rack their brains to explore new fraud strategies and make a fraud transaction as similar to a genuine one in order to avoid being found by a fraud detection system. Although fraudsters try to behave like the real cardholders, they have no way to know the real spending habit of cardholders, and they are eager to transfer all money to other accounts. Hence, these different transaction behaviors or psychological activities lead to different features between fraud and genuine transaction data. Fraud strategies are maybe changed, but the purpose of fraudsters will never change. Therefore, it is very significant to extract effective representations that can steadily distinguish fraud transactions from genuine ones even when fraud strategies are often changed.

In this article, our objective is to build a credit card fraud detection model based on deep representation learning methods that can learn effective representations of transaction behaviors. Simultaneously, we hope that our model can have good stability. For the class imbalance problem, there are many methods to handle it. This article pays more attention to a better learning representation that can both enhance the performance of fraud detection and keep the stability of performance. As mentioned in the literature [12], a representation learning method is to learn representations of the data that can easily extract useful information when building classifiers or other predictors. Representation learning has been applied widely such as person reidentification [13] and face recognition [14].

Manuscript received November 13, 2019; revised January 14, 2020; accepted January 20, 2020. Date of publication February 17, 2020; date of current version April 3, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB2100801 and in part by the Fundamental Research Funds for the Central Universities of China under Grant 22120190198. *(Corresponding authors: Guanjun Liu; Changjun Jiang.)*

The authors are with the Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Department of Computer Science, Tongji University, Shanghai 201804, China, and also with the Shanghai Electronic Transactions and Information Service Collaborative Innovation Center, Tongji University, Shanghai 201804, China (e-mail: 1510482@tongji.edu.cn; liuguanjun@tongji.edu.cn; cjjiang@tongji.edu.cn).

Digital Object Identifier 10.1109/TCSS.2020.2970805

We present a deep neural network as our representation learning model, mapping the original features of transactions into deep representations for identifying fraud transactions accurately. Intuitively, the learned deep representations should maximize their intraclass compactness and interclass separability simultaneously. We construct a novel function full center loss (FCL) as the loss function of our deep representation learning model. FCL integrates two different aspects of optimization objectives (or losses). The first aspect is about the distance between deep representation and class center, named distance center loss (DCL). DCL can stress the intraclass compactness. The second aspect is an optimized softmax loss (SL) with a maximum angle that can promote the deep representations of samples from different classes so that the interclass separability is improved. Since the SL can change the angular distribution of learned representations [15], we call our optimized SL as angle center loss (ACL). To demonstrate a stronger feature leaning ability of our model, we provide experimental results by comparing it with state-of-the-art methods on different data sets. Our contributions are summarized as follows.

1) A novel loss function, ACL, is proposed to improve the separability of learned features. ACL is an optimized SL function by addressing the problem of maximum angle separation.
2) We construct a novel FCL by integrating ACL and DCL. FCL ensures a stronger ability to mapping original transaction features into much more distinguishing deep representations since it fully considers the distance and angle of features of transactions.
3) We summarize the state-of-the-art loss functions used in the deep representation learning methods and compare them with ours over two big data sets. We also demonstrate that our model has better performance stability.

The rest of this article is organized as follows. In Section II, we review the related work. Section III presents the framework of our credit card fraud detection model. Section IV describes our loss functions. Then, the details of our experiments are shown in Section V. Section VI presents a conclusion of this article and some future studies.

II. Related Work

A. Fraud Detection Model

Credit card fraud detection is one of the most popular topics in the fraud detection fields, especially with the sustainable growth of e-commerce transactions in recent years. Generally, fraud detection is very challenging because of two major problems: class imbalance and data dynamic change [8].

The class imbalance problem of credit card fraud detection has been studied for a long time [16]. One of the most famous methods is resampling [9]. Concretely, a training data set can be balanced by removing some samples from the majority class (i.e., undersampling [17]) or generating some samples for the minority class (i.e., oversampling [10]). In addition, ensemble methods [18], e.g., bagging, boosting, and stacking, are also often used to solve the class imbalance problem. Cost-sensitive learning [19] is another way to deal with it through assigning different misclassification error costs for different classes, and a higher cost is assigned to a minority class generally. Apart from the quantity imbalance, the spatial distribution of instances from different classes is also an important factor that influences the result of classifiers. For example, the samples nearby the cross edge of majority and minority are more important for a classifier since they are harder to be identified accurately. Therefore, the Gaussian mixture undersampling method [20] is proposed to sample more informative instances and, thus, improve the performance of classifiers.

The reason leading to the problem of transaction data dynamic change is the changing and evolving of users' transaction behaviors. The changes in consumption seasonality and fraud patterns lead to the deformation of the distribution of the original transaction data. However, the commonly used fraud detection methods, such as support vector machine (SVM) [21], random forests (RF) [22], and convolutional neural networks (CNNs) [4], generally assume that classes are balanced and data distribution is unchanged. Literature [23] presents a very comprehensive survey of methods of credit card fraud detection from 1990 to 2017. Most of these methods focus on combining different class imbalance processing methods to improve the performance of a classifier but do not consider the data dynamic change problem.

A few studies consider the data dynamic change problem as the concept drift [5], [24]. They mainly focus on timely detecting the appearance of concept drift and adaptively updating a classifier to fit for new concepts. As discussed in Section I, there exist inherent difference features between fraud and genuine transactions. Therefore, it is also very significant to extract effective representations that can steadily distinguish fraud transactions from genuine ones even when fraud strategies are often changed. In this article, we focus on design a good and stable fraud detection model based on deep representation learning methods [12].

B. Deep Supervised Representation Learning

A representation learning method is to learn another kind of representation for the given data, and it makes this new representation capture more useful information that can be used to build a better classifier or predictor [12]. It has achieved great success in many domains, especially in large-scale visual classification with supervised learning settings to extract discriminative features [25]. Due to the existence of many public visual data sets, such as ImageNet [26], LFW [27], and COCO [28], many studies about deep representation learning are explored and applied in the visual domain.

The architecture of a deep neural network influences the performance ceiling of the corresponding representation model. As pointed out in [25], the capacity of CNNs can be controlled by varying their depth and breadth. They also make some strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Many efforts have been put into the study of architectures of CNNs and achieved remarkable success, such as ResNet [29], DenseNet [30], and BagNet [31].

Apart from network architecture, the loss function is another key factor because it directly determines what a representation learning model can be achieved. The research about loss functions of deep representation learning models has also attracted much attention. According to the number of instances required in a loss calculation, loss functions can be roughly divided into three kinds: individual-sample-based SL functions [32]–[35], sample-pair-based contrastive loss functions [36], [37], and sample-triplet-based triplet loss functions [38]–[40].

The triplet loss [38] considers triplets of samples (x_0, x_+, x_-), where (x_0, x_+) with the same label forms a positive pair and (x_0, x_-) with different labels forms a negative pair. It is used to measure the difference of their distances or similarities and encourages those learned representations with the same label to be closer than those with different labels. There are some variants of triplet loss [39], [40]. However, mining hard triplets of samples is too time-consuming for some tasks with large data sets and suffers from the problem of dramatic data expansion. In order to deal with them, a batch-hard-based triplet loss is introduced in [39]. It can generate hard negative and hard positive sample pairs from every training batch. In addition, [40] proposes a triplet center loss that replaces x_+ and x_- with the corresponding class center and another noncorresponding class center. Thus, it can make the learned representations closer to their corresponding class center and further away from other class centers.

The contrastive loss [36] is to minimize the distances of positive sample pairs and maximize the distances of negative sample pairs if the distances are not larger than some preset margin. Just like the triplet loss, the contrastive loss also suffers from the problem of time-consuming of preparing the sample pairs. Wen *et al.* [15] propose a center loss to deal with this problem. The center loss learns a center for the deep representations of every class and minimizes the distances between the learned representations and their corresponding class centers. Although the time complexity of preparing for sample pairs can be reduced, the contrastive loss cannot directly be applied to a classification task because it is only able to supervise a model to learn discriminative representations. Hence, it is often combined with SL as an auxiliary objective to learn separable and discriminative representations [15], [37].

The SL is widely used in deep representation learning methods due to its simplicity and effectiveness. However, recent studies [32]–[34] illustrate that the conventional SL is ineffective to reduce the intraclass variation. To handle this problem, some studies [15], [37] directly combine the contrastive loss with it. Other studies focus on enhancing its discrimination ability. In addition, some studies are to design different forms of angle margin between learned representations and their target weights. For example, Liu *et al.* [32] propose a large-margin SL (LMSL) in terms of angular similarity. They multiply a preset constant m with the angle between a sample's feature vector and the weight vector of its corresponding class for a potentially larger angular separability among those learned deep representations. They also propose an angular SL (ASL) [33] that puts more constraints on the weights from the last fully connected layer to directly optimize angles. This can learn those angularly distributed representations with angular margin. Reference [34] presents a large-margin cosine loss (LMCL) that directly uses cosine margin of different classes. This can improve the cosine-related discriminative information and can be implemented easily. Recently, Deng *et al.* [35] propose an additive angular margin loss (AAML) that adopts the arccosine function to calculate the angle between a learned representation and the target weight so that it can optimize the geodesic distance margin under an exact correspondence from angle to arc in the normalized hypersphere.

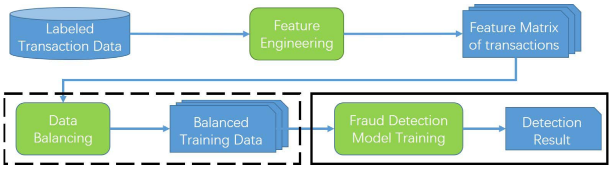


Fig. 1. Process of building a credit card fraud detection model.

During the process of training a model with SL, the learned representation $\boldsymbol{f}_i$ of a sample $\boldsymbol{x}_i$ with label 0 (respectively 1) is pulled closer to its target weight $\boldsymbol{W}_0$ (respectively $\boldsymbol{W}_1$),[1] and vice versa. The purpose of angle margin is to ensure a smaller average angle between $\boldsymbol{F}_0$ (respectively $\boldsymbol{F}_1$)[2] and $\boldsymbol{W}_0$ (respectively $\boldsymbol{W}_1$) through increasing the punishment to the same angle compared with the conventional SL. However, it has not been fully explored how to enhance the separability of the learned representations via SL. Therefore, for addressing this problem, this article proposes a novel ACL. ACL is an improved SL function with the maximum angular separation. It can make the learned representations of different classes separated in opposite directions. Hence, it can directly obtain the optimal separability of learned representations. More details are shown in Section IV.

III. Credit Card Fraud Detection Model

As shown in Fig. 1, building an effective credit card fraud detection model consists of some essential steps which significantly influence the detection.

A. Feature Engineering

The first step is feature engineering that is aiming at extracting informative features of users' transaction behaviors. The raw features, such as transaction time/date and transaction amount, cannot well characterize the transaction behaviors of cardholders and fraudsters. One of the commonly used methods is to derive some new features using the transaction aggregation strategy [41]. The aggregation features are derived through grouping the transactions according to a selected time interval, card number, transaction type, and merchant code. Then, the number of transactions and the total amount spent on those transactions are calculated. After the process of transaction aggregation strategy, a single transaction with

[1] $\boldsymbol{W}_0$ (respectively $\boldsymbol{W}_1$) is the weight of the class labeled by 0 (respectively 1).

[2] $\boldsymbol{F}_0$ (respectively $\boldsymbol{F}_1$) is the set of the learned representations corresponding to those samples with label 0 (respectively 1).

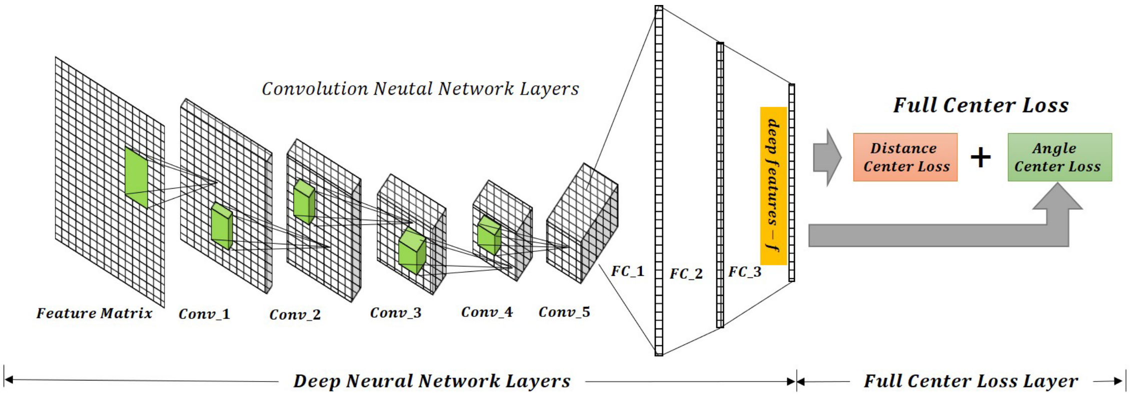


Fig. 2. Our deep representation learning framework consisting of two parts: the deep neural network layers (e.g., convolution neural network layers) and the FCL layers.

raw features is transformed into a feature matrix with more informative aggregation features. The transaction aggregation strategy has been adopted in many studies [4], [42]. Our previous work [42] also shows that it can well distinguish the transaction behaviors of cardholders and fraudsters. In this article, we also use this strategy.

B. Data Balancing

After feature engineering, a classifier can be trained as a binary classification task. However, if the class imbalance problem is not considered, the learned classifier will tend to identify most of the fraud transactions as genuine ones. The reason is that almost all classifiers have a default assumption of a balanced data set, and thus, the learned decision boundary tends to bias toward the class with more samples. Hence, dealing with the class imbalance problem has become an indispensable step before training a fraud detection model. As mentioned in Section II-A, the most commonly used method of handling the class imbalance problem is data sampling. Especially, the undersampling method can reduce the redundancy of genuine transactions and speed up the model training. Randomly undersampling is one of the most famous undersampling methods due to its simplicity and effectiveness. However, these sampling methods do not consider the spatial distribution of instances from different classes. The Gaussian mixture undersampling method [20] can be applied to sample more informative instances and, thus, improve the performance of classifiers. However, if a data set has quite a few fraud transactions compared with the legal ones, the upsampling method, such as SMOTE [10], should be applied to enrich the fraud transactions. In this article, we apply different data balancing methods according to different data sets that will be introduced in Section V.

C. Fraud Detection Model Training

A fraud transaction detection model, as a binary classifier, can be trained with a relatively balanced data set after handling the class unbalance problem. There are many machine learning methods, such as SVM [21], RF [22], CNNs [4], and recurrent neural networks (RNNs) [6], that have been successfully used to detect fraud transactions. Almost all of them belong to representation learning. They aim at discovering better representations of inputs by learning transformations of data that disentangle factors of variation in data and retain most of the information [12]. Especially, the deep representation learning with deep neural networks has achieved remarkable success in many domains in recent years due to some advanced structures, such as ResNet [29], DenseNet [30], and BagNet [31], and some effective loss functions, such as AAML [35], center loss [15], and triplet loss [38]. The latest neural network architectures make a deep representation learning model not only deeper with much more layers but also easier for model training. These advanced architectures significantly enhance the ability of complex nonlinear mapping of a deep representation learning model. On the other hand, the ingeniously designed loss functions can supervise the process of deep representation so that the final model can obtain an ideal result.

As shown in Fig. 2, the fraud transaction detection model adopted in this article is composed of two parts: the deep neural network layers (e.g., convolution neural network layers) for obtaining separable and discriminative representations, and the fully center loss layer for supervising the model training. The key of this article is to optimize the loss function so that the quality of those learned deep features and the performance of fraud transaction detection are enhanced. Our loss function is to supervise the training of deep convolution neural network layers that project the original feature space of transactions into a deep feature space. The goal is that the transactions from the same class can be compact as fully as possible and the transactions from different classes can be separated as fully as possible. For this goal, we specially design our FCL, combining two different aspects of losses: ACL for addressing the separability of transactions from different classes and DCL for addressing the compactness of transactions from the same class.

IV. FULL CENTER LOSS (FCL)

ACL is an improved SL, which is specially designed for our credit card fraud detection as a binary classification problem. It can enhance the classification ability compared to the softmax function. Meanwhile, the advantage of the simplicity of the SL is still retained in ACL. DCL is originally proposed in [15] and is to measure the aggregation of deep features of each class.

FCL can be formulated as follows:

$$L_{\text{Full}} = \sum_{i=1}^{m} (L_{A\_i} + \alpha L_{D\_i}) \tag{1}$$

where α is a hyperparameter to trade off these two losses and m denotes the size of minibatch samples for training our deep representation learning model. The detailed explanation of every loss is shown in the following.

A. Angle Center Loss (ACL)

In general, the ideal deep features should keep intraclass compactness and interclass separability as much as possible. Although the SL of the normal CNN model is very simple and performs well in many classification applications, it is not too effective to generate discriminative features. When using the original SL to solve a binary class problem, the posterior probabilities of the learned deep representation $\boldsymbol{f}_i$ of an input sample $\boldsymbol{x}_i$ with label 0 or 1 can be written as

$$p_0 = \frac{e^{\boldsymbol{W}_0^T \boldsymbol{f}_i + b_0}}{e^{\boldsymbol{W}_0^T \boldsymbol{f}_i + b_0} + e^{\boldsymbol{W}_1^T \boldsymbol{f}_i + b_1}} \tag{2}$$

$$p_1 = \frac{e^{\boldsymbol{W}_1^T \boldsymbol{f}_i + b_1}}{e^{\boldsymbol{W}_0^T \boldsymbol{f}_i + b_0} + e^{\boldsymbol{W}_1^T \boldsymbol{f}_i + b_1}} \tag{3}$$

where ($\boldsymbol{W}_0$, b_0) and ($\boldsymbol{W}_1$, b_1) are weights and bias of the softmax layer in CNN corresponding to class 0 and 1, respectively. $\boldsymbol{f}_i$ is the output of the last fully connected layer. p_0 and p_1 are the posterior probabilities of $\boldsymbol{f}_i$ belonging to classes 0 and 1, respectively.

It has been verified that features learned by the original SL have intrinsic angular distribution [33]. If an input deep representation $\boldsymbol{f}_i$ has the label y_i, then the original SL of deep representation $\boldsymbol{f}_i$ can be reformulated as follows:

$$\begin{aligned} L_{\text{softmax\_i}} &= -\log\left(\frac{e^{\boldsymbol{W}_{y_i}^T \boldsymbol{f}_i + b_{y_i}}}{e^{\boldsymbol{W}_{y_i}^T \boldsymbol{f}_i + b_{y_i}} + e^{\boldsymbol{W}_{\tilde{y}_i}^T \boldsymbol{f}_i + b_{\tilde{y}_i}}}\right) \\ &= -\log\left(\frac{1}{1 + e^{(\boldsymbol{W}_{\tilde{y}_i}^T - \boldsymbol{W}_{y_i}^T)\boldsymbol{f}_i + (b_{\tilde{y}_i} - b_{y_i})}}\right) \end{aligned} \tag{4}$$

where $\tilde{y}_i$ denotes another class different from y_i in the binary classification.

We design two constraints for the original SL to keep the angular separability of instances from different classes. First, following the modified SL in [33], we normalize $\|\boldsymbol{W}\| = \|\boldsymbol{W}_{y_i}\| = \|\boldsymbol{W}_{\tilde{y}_i}\| = 1$ and set the biases $\|b_{y_i}\| = \|b_{\tilde{y}_i}\| = 0$ to maintain the angular boundary. This can guarantee that the value of SL just depends on the norm value of deep representations $\boldsymbol{f}_i$ and the angles between $\boldsymbol{W}_j$ and $\boldsymbol{f}_i$. Therefore, the modified SL of deep feature vector $\boldsymbol{f}_i$ can be written as

$$\begin{aligned} L_{\text{modified}\_i} &= -\log\left(\frac{e^{\|\boldsymbol{f}_i\| \cos\theta_{y_i}}}{e^{\|\boldsymbol{f}_i\| \cos\theta_{y_i}} + e^{\|\boldsymbol{f}_i\| \cos\theta_{\tilde{y}_i}}}\right) \\ &= -\log\left(\frac{1}{1 + e^{-\|\boldsymbol{f}_i\|(\cos\theta_{y_i} - \cos\theta_{\tilde{y}_i})}}\right) \end{aligned} \tag{5}$$

where $0 \le \theta_{y_i}, \theta_{\tilde{y}_i} < 2\pi$, θ_{y_i} denotes the angle between vectors $\boldsymbol{W}_{y_i}$ and $\boldsymbol{f}_i$, and $\theta_{\tilde{y}_i}$ denotes the angle between vectors $\boldsymbol{W}_{\tilde{y}_i}$ and $\boldsymbol{f}_i$.

In order to minimize the value of L_{modified_i}, intuitively, two optimizations should be taken into account. On the one hand, θ_{y_i} should be decreased, and $\theta_{\tilde{y}_i}$ should be increased. On the other hand, the value of $\|\boldsymbol{f}_i\|$ should be enlarged since we have already constrained $\|\boldsymbol{W}\| = 1$. Considering the fact that the value range of $\|\boldsymbol{f}_i\|$ is closely related to the DCL, we put more focus on the feasible optimization measure to make θ_{y_i} smaller than $\theta_{\tilde{y}_i}$ as much as possible. Ideally, instances from the same class are closely distributed on both sides of their corresponding $\boldsymbol{W}_{y_i}$. Hence, $\boldsymbol{W}_{y_i}$ can be regarded as the angle center of each class instance, and thus, we name this loss as ACL.

To obtain a stable classification performance, the learned deep representations of instances from different classes should keep separability as fully as possible. However, the modified SL is only able to directly minimize the angles between $\boldsymbol{f}_i$ and its corresponding $\boldsymbol{W}_{y_i}$. Therefore, we design another stronger constraint, i.e., $\boldsymbol{W}_{y_i}$ and $\boldsymbol{W}_{\tilde{y}_i}$ are in opposite directions: $\boldsymbol{W} = \boldsymbol{W}_{y_i} = -\boldsymbol{W}_{\tilde{y}_i}$. Finally, ACL with deep representation $\boldsymbol{f}_i$ can be reformulated as

$$\begin{aligned} L_{A\_i} &= -\log\left(\frac{1}{1 + e^{-2\|\boldsymbol{f}_i\| \cos\theta_{y_i}}}\right) \\ &= -\log\left(\frac{1}{1 + e^{-2\boldsymbol{W}^T \boldsymbol{f}_i}}\right) \\ &= \log\left(1 + e^{-2\boldsymbol{W}^T \boldsymbol{f}_i}\right). \end{aligned} \tag{6}$$

From (6), it is easy to infer that with the decreasing of ACL, the deep representation $\boldsymbol{f}_i$ will gradually get close to $\boldsymbol{W}_{y_i}$, and thus, the learned representations from different classes will gradually be separated in the opposite directions since $\boldsymbol{W}_{y_i}$ and $\boldsymbol{W}_{\tilde{y}_i}$ are always in the opposite directions.

Though we apply these stronger constraints to the original SL, ACL still remains the superiorities of the original SL. Therefore, it can still be easily optimized with gradient computation and backpropagation. When we replace $\cos\theta_{y_i}$ with $\boldsymbol{W}$ and $\boldsymbol{f}_i$, the derivative of ACL with respect to $\boldsymbol{W}$ and $\boldsymbol{f}_i$ can still be easily computed, just like the original SL.

B. Distance Center Loss (DCL)

DCL is mainly responsible for the separability of the learned deep representations from different classes. As for the compactness of intraclass deep representations, we adopt the center loss [15] in Euclidean space, which can be measured by the distance from an instance to its corresponding center. The center loss of deep representation $\boldsymbol{f}_i$ can be formulated as follows:

$$L_{D\_i} = \frac{1}{2}\|\boldsymbol{f}_i - \boldsymbol{c}_{y_i}\|_2^2 \tag{7}$$

where $\boldsymbol{c}_{y_i}$ denotes the corresponding class center of $\boldsymbol{f}_i$ with label y_i.

As the original center loss directly uses the Euclidean distance, we name it as DCL. It has been proven that CNNs supervised by DCL are trainable and can be optimized through

TABLE I

DIFFERENT METHODS FOR IMPROVING SL

Loss Functions	Key Formulas
Softmax Loss (SL)	$(\boldsymbol{W}_{\tilde{y_i}}^T - \boldsymbol{W}_{y_i}^T)\boldsymbol{f_i} + (b_{\tilde{y_i}} - b_{y_i})$
Large Margin Softmax Loss (LMSL) [32]	$\|\boldsymbol{f_i}\|(\|\boldsymbol{W}_{\tilde{y_i}}\| \cos\theta_{\tilde{y_i}} - \|\boldsymbol{W}_{y_i}\| \cos m\theta_{y_i})$
Angular Softmax Loss (ASL) [33]	$\|\boldsymbol{f_i}\|(\cos\theta_{\tilde{y_i}} - \cos m\theta_{y_i})$
Large Margin Cosine Loss (LMCL) [34]	$s(\cos\theta_{\tilde{y_i}} - (\cos(\theta_{y_i}) - m))$
Additive Angular Margin Loss (AAML) [35]	$s(\cos\theta_{\tilde{y_i}} - \cos(\theta_{y_i} + m))$
Angle Center Loss (ACL)	$-2\|\boldsymbol{f_i}\| \cos\theta_{y_i}$

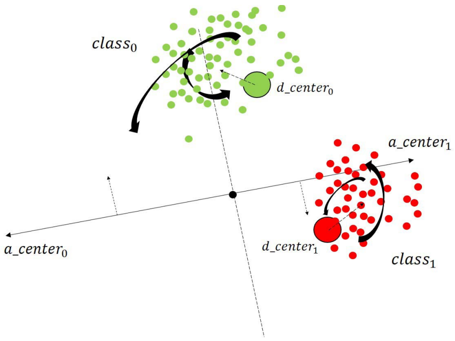


Fig. 3. Idea of our FCL that consists of ACL and DCL.

the standard stochastic gradient descent (SGD) method [15]. When training a model, the distance center of every class is computed by averaging the learned deep representations in the class in Euclidean space. A scalar γ is adopted to limit the updating speed of the distance centers to avoid large perturbations caused by a few mislabeled samples. Therefore, following the definition in article [15], the update equation of $\boldsymbol{c}_{y_i}$ can be formulated as:

$$\frac{\partial L_{D\_i}}{\partial \boldsymbol{f}_i} = \boldsymbol{f}_i - \boldsymbol{c}_{y_i} \tag{8}$$

$$\Delta \boldsymbol{c}_j = \frac{\sum_i^m \delta(y_i = j) \cdot (\boldsymbol{c}_j - \boldsymbol{f}_i)}{1 + \sum_i^m \delta(y_i = j)} \tag{9}$$

$$\boldsymbol{c}'_j = \boldsymbol{c}_j - \gamma \cdot \Delta \boldsymbol{c}_j \tag{10}$$

where $\delta(\text{condition}) = 1$ if the condition is satisfied, otherwise $\delta(\text{condition}) = 0$. $\boldsymbol{c}'_j$ is the center of class j after updating, and the hyperparameter γ is limited in range [0, 1].

Combing ACL and DCL, the learning details of our proposed fraud detection model can be summarized as in Algorithm 1. Fig. 3 shows the idea about FCL visually. a_center_0 and a_center_1 denote the centers of classes 0 and 1 in angel space, respectively. a_center_0 and a_center_1 are always in two opposite directions for the best angle separation. d_center_0 and d_center_1 denote the centers of classes 0 and 1 in Euclidean space, respectively. When training a model, each training sample and the related centers (distance center and angle center) will be pulled closer gradually.

C. Some State-of-the-Art Loss Functions

In this section, we introduce some state-of-the-art loss functions based on SL. They can improve the feature learning ability of the representation learning model compared with the original SL. We compare them with ours (including ACL and FCL) in our experiments, which is shown in Table I.

Algorithm 1 Deep Representations Learning With FCL

```
Require: Training data set of transactions $X = \{\boldsymbol{x}_i\}(i \in \{1, 2, \ldots, m\})$ and the learned deep representations $F = \{\boldsymbol{f}_i\}$ corresponding to $X$. Initialized parameters $\boldsymbol{\theta}_c$ in convolution-based deep representation learning layers. The initialized parameters $\boldsymbol{W}$ and $\{\boldsymbol{c}_k\}(k = 0, 1)$ of angle center loss and distance center loss. Hyperparameters $\alpha$, $\gamma$ and the learning rate $\mu_t$. The number of iterations $t$ is initialized by 0.
Ensure: The learned parameters $\boldsymbol{\theta}_c$, $\boldsymbol{W}$ and $\boldsymbol{c}_k$.
repeat
    $t = t + 1$;
    Compute FCL $L_{Full}^t = \sum_{i=1}^m L_{A\_i}^t + \alpha L_{D\_i}^t$;
    Compute the backpropagation error $\frac{\partial L_{Full}^t}{\partial \boldsymbol{f}_i^t} = \frac{\partial L_{A\_i}^t}{\partial \boldsymbol{f}_i^t} + \alpha \frac{\partial L_{D\_i}^t}{\partial \boldsymbol{f}_i^t}$;
    Update $\boldsymbol{W}$: $\boldsymbol{W}^{t+1} = \boldsymbol{W}^t - \mu_t \frac{\partial L_{Full}^t}{\partial \boldsymbol{f}_i^t} = \boldsymbol{W}^t - \mu_t \frac{\partial L_{A\_i}^t}{\partial \boldsymbol{f}_i^t}$;
    Update $\boldsymbol{c}_k$: $\boldsymbol{c}_k^{t+1} = \boldsymbol{c}_k^t - \gamma \Delta \boldsymbol{c}_k^t$;
    Update $\boldsymbol{\theta}_c$: $\boldsymbol{\theta}_c^{t+1} = \boldsymbol{\theta}_c^t - \mu_t \sum_i^N \frac{\partial L_{Full}^t}{\partial \boldsymbol{f}_i^t} \frac{\partial \boldsymbol{f}_i^t}{\partial \boldsymbol{\theta}_c^t}$;
until Parameters converge
```

The LMSL [32] multiplies a preset constant m with the angle θ between the learned representation $\boldsymbol{f}_i$ of a sample and the weight vector $\boldsymbol{W}_{y_i}$ of its corresponding class. Therefore, under the same $\boldsymbol{f}_i$ and $\boldsymbol{W}_{y_i}$, more punishments are attached to the representation learning model with LMSL than that with SL. Therefore, potentially larger angular separability of the learned deep representations can be achieved. However, the influence from the norm value of weight $\boldsymbol{W}_{y_i}$ is ignored in LMSL, counteracting the work of parameter m.

The ASL [33] addresses the above-mentioned problem of LMSL and puts more constraints on the weights to make the SL directly optimize angles for learning angularly distributed representations with an angular margin by normalizing $\|\boldsymbol{W}_{y_i}\| = 1$. However, a complicated calculation of ASL limits the speed of model training.

The LMCL [34] is easily implemented since it directly adds a cosine margin between two different classes so that it can improve the cosine-related discriminative information. $\boldsymbol{W}_{y_i}$ is normalized as $\|\boldsymbol{W}_{y_i}\| = 1$, and the learned representation $\boldsymbol{f}_i$ is also fixed as $\|\boldsymbol{f}_i\| = s$ for simplifying the calculation of cosine similarity. However, $\|\boldsymbol{f}_i\| = s$ is just appropriate to the data with a limited variety, such as the face images. For credit card transaction data, because the genuine transactions are diversiform, uniformizing them will increase the difficulty of model learning.

The AAML [35] adds an additive angular margin penalty m between $\boldsymbol{f}_i$ and $\boldsymbol{W}_{y_i}$ in order to simultaneously enhance the intraclass compactness and interclass discrepancy. Just as LMCL, $\boldsymbol{W}_{y_i}$ is normalized as $\|\boldsymbol{W}_{y_i}\| = 1$, and the learned representation $\boldsymbol{f}_i$ is also fixed as $\|\boldsymbol{f}_i\| = s$ in AAML. AAML can be easily calculated by utilizing the arccosine function.

In our ACL, $\boldsymbol{W}_{y_i}$ is also normalized as $\|\boldsymbol{W}_{y_i}\| = 1$, but the deep representation $\boldsymbol{f}_i$ is learned since it can influence the performance of a softmax-based classifier. All of the above-mentioned methods introduce a fix margin m to address the angular separation. They introduce punishment to pull $\boldsymbol{f}_i$ close to $\boldsymbol{W}_{y_i}$ and push $\boldsymbol{f}_i$ away from $\boldsymbol{W}_{\tilde{y}_i}$. Therefore, $\boldsymbol{f}_i$ will get closer to $\boldsymbol{W}_{y_i}$ with the increasing of the training epoch. Therefore, the orientation of $\boldsymbol{W}_{y_i}$ can be regarded as the angular center of learned representations. The norm of $\boldsymbol{W}_{y_i}$ has been fixed as 1 for better angular separation, but the influence from the orientation of $\boldsymbol{W}_{y_i}$ and $\boldsymbol{W}_{\tilde{y}_i}$ has not been fully excavated. Therefore, in our ACL, the orientations of $\boldsymbol{W}_{y_i}$ and $\boldsymbol{W}_{\tilde{y}_i}$ are set oppositely for the largest angular separation of $\boldsymbol{f}_i$ from different classes. Based on this setting, the number of parameters of ACL is only half of that of SL.

V. Experiments

A. Data Sets

The first data set of credit card fraud detection is from Kaggle [43] that is public and used in many studies. The other one is our private transaction data set from a financial company in China.

1) Data Set From Kaggle: This data set is composed of credit card transactions of European cardholders in September 2013. These transactions are generated in two days, and there are 492 fraud transactions out of all 284 807 transactions. It is obvious that this data set is very imbalanced: the positive class (fraud instances) only accounts for 0.172% of all transactions. Every transaction has 30 features (Time, $V1$, $V2, \ldots, V28$, Amount) in this data set. Except for Time and Amount, all other features are numerical values that are generated from the original features with a PCA transformation. For the confidentiality issues, the original features and details about this data set are not provided. Because all transactions in this data set cannot be associated with the information of corresponding cardholders, it is unpractical to aggregate features for acquiring more useful information. Hence, this data set with original features is directly applied to train and test models. In other words, the steps of feature engineering and feature matrix in Fig. 1 are not used for this data set.

2) Our Private Data Set: This data set contains fraud and genuine transactions labeled by professional investigators of the company. There are up to 3.5 million transactions in this data set from April to June in 2017. Table II shows the details of the transactions in each month, and the class imbalance problem is also very serious. The original features of each transaction include transaction time, transaction amount, transaction type, merchant type, currency type, and card type. Literature [8] underlines that features extracted from a single transaction are much less informative or sufficient to detect a fraud occurrence than the aggregate features combined with historical transactions. Therefore, we adopt some aggregation measures on original features in our previous work [42]. The original features are aggregated with different time intervals from 5 s to two months and extended according to different feature types. Table III shows the example of the feature matrix after the original features are aggregated. For more details about the feature aggregation, one can refer to [42]. After the process of feature aggregation, every transaction is transformed into an informative feature matrix that represents more comprehensive information of this transaction and the spending habits of the corresponding cardholder. These informative feature matrices are adopted for model training and testing.

TABLE II
Experiment Data Set

Date	Instances	Original Features	Fraud Rate
2017-04	1,243,035	43	1.07%
2017-05	1,216,299	43	2.22%
2017-06	1,042,714	43	2.39%

TABLE III
Aggregated Feature Matrix

Time Interval Features Names	...	1h	1d	7d	...
Avg_Amount					
Total_Amount					
Max_Amount					
Most_Checking_Method					
Gini_Change_MAC					
...					

B. Models

To show the advantages of our fraud detection model with FCL, the state-of-the-art loss functions in Table I are compared with ours. The RF model is one of the most popular models for detecting fraud transactions and is also compared with the benchmark in this article.

Since the data set from Kaggle has no more than 0.3 million transactions with 30 usable features, we build an artificial neural network with four fully connected layers for learning effective representations. Because our private data set has millions of transactions with informative feature matrices, a more powerful CNN is adopted to learn deep representations. We use five convolution layers with max-pooling and three fully connected layers following the last convolution layer. More details about the parameters of these neural network structures are shown in Table IV.

For the implementation of different loss functions, we utilize the online code repositories and modify them according to the requirement of credit card fraud detection. The best hyperparameters of each loss function are searched by the grid search method.

TABLE IV
KEY HYPERPARAMETERS OF EACH MODEL FOR DIFFERENT DATA SETS

Data Models	Kaggle Dataset	Financial company Dataset

Random Forests	tree_num=p
	max_deep=q
	p,q with grid search
		tree_num=p
		max_deep=q
		p,q with grid search

Neural Networks	ANN based model with 3 fully connected layers: FC-1:32, FC-2:64, FC-3:16, FC-4: 8	CNN based model with 5 convolutional layers and 3 fully connected layers: Conv-1: $[1 \times 1, 32]$ S1, Conv-2: $\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix}$ S1, MaxPool-1: [3 × 3] S2, Conv-3: $\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix}$ S1, MaxPool-2: [3 × 3] S2, FC-1: 256, FC-2: 128, FC-3: 64

TABLE V
CONFUSION MATRIX

	True Fraud	True Genuine
Predicted Fraud	TP	FP
Predicted Genuine	FN	TN

C. Performance Measures

The conventional confusion matrix of binary classification is shown in Table V.

The commonly used performance measures are accuracy, precision, and recall. However, they are not enough for measuring model performance due to the class imbalance problem. F_β score [44] is the weighted harmonic mean of precision and recall that can measure model performance comprehensively. β is commonly 1 (i.e., F_1) to equally treat precision and recall. Area under precision–recall curve (AUC_PR) [45] is another very appropriate metric for evaluating model performance with imbalanced data sets because of its susceptibility of classifiers to imbalanced data sets

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} y \tag{11}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

$$F_\beta = \frac{(1 + \beta^2) \times \text{recall} \times \text{precision}}{\text{recall} + \beta^2 \times \text{precision}} \tag{14}$$

$$F_1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}. \tag{15}$$

D. Experiment I

The first experiment is conducted on the Kaggle data set. Considering there are only 492 fraud transactions in the data set that can lead to a big variance of the test results for different models, we use the upsampling method SMOTE [10] to generate more fraud transactions according to the actual transactions to balance the data set with ratio 1:1. Then, 70% of transactions are randomly selected as the training set and the rest as the test set. The best hyperparameters m, s, α, and γ in different loss functions are searched with the grid search method.

TABLE VI
PERFORMANCE OF DIFFERENT MODELS ON KAGGLE DATA SET

Method	F_1	*AUC_PR*
RF	0.378	0.564
SL	0.683	0.804
LMSL	0.631	0.828
ASL	0.729	0.808
LMCL	0.671	0.817
AAML	0.713	0.839
ACL	**0.736**	**0.848**
FCL(ACL+DCL)	**0.805**	**0.879**

TABLE VII
PERFORMANCE ON DATA SET FROM FINANCIAL COMPANY

Method	F_1	*AUC_PR*
RF	0.787	0.773
SL	0.791	0.782
LMSL	0.796	0.803
ASL	0.804	0.788
LMCL	0.798	0.806
AAML	0.801	0.797
ACL	**0.807**	**0.811**
FCL(ACL+DCL)	**0.813**	**0.825**

Table VI shows the average results of F_1 and AUC_PR. First, the performance of every neural network model is better than RF because of the nonlinear characteristic combination ability of a neural network. Second, our ACL outperforms other loss functions, which means that the maximum angle separation in our ACL is superior to others. In addition, when combining ACL with DCL, the performance of our models (i.e., FCL) is continuously improved obviously, which indicates the importance of intraclass compactness of learned representations. This also shows that FCL can strengthen the deep representation learning models to obtain better representations and improve fraud detection performance.

E. Experiment II

This experiment is conducted on our private data set from a financial company. This data set has abundant transactions over three consecutive months, but the class imbalance problem is still serious. Our previous work, Gaussian mixture undersampling [20] method, can sample informative instances to handle the class imbalance problem, and thus, it is still adopted in this experiment. Since every original feature of transactions has a concrete meaning, every transaction can be transformed into an informative feature matrix by the feature aggregation method [42].

This experiment consists of two subexperiments. The first one is to verify the advantage of our ACL and FCL, and the

TABLE VIII

PERFORMANCE CHANGE WITH DIFFERENT TEST SET

training data (4.1–4.30)												
Method	T_1*(5.1–5.10)*		T_2*(5.11–5.20)*		T_3*(5.21–5.31)*		T_4*(6.1–6.10)*		T_5*(6.11–6.20)*		T_6*(6.21–6.30)*	
	F_1	*AUC_PR*	F_1	*AUC_PR*	F_1	*AUC_PR*	F_1	*AUC_PR*	F_1	*AUC_PR*	F_1	*AUC_PR*
SL	0.756	0.764	0.794	0.790	0.644	0.524	0.676	0.588	0.748	0.714	0.706	0.642
LMSL	0.788	0.782	0.805	0.834	0.669	0.539	0.682	0.612	0.738	0.785	0.715	0.636
ASL	0.792	0.801	0.826	0.829	0.679	0.571	0.694	0.645	0.754	0.782	0.738	0.699
LMCL	0.799	0.820	0.819	**0.863**	0.696	0.627	0.705	0.690	0.776	0.810	0.727	0.691
AAML	0.816	0.817	0.825	0.834	0.722	0.675	0.755	0.701	0.799	0.814	0.736	0.725
ACL	**0.820**	**0.835**	**0.843**	0.858	**0.751**	**0.701**	**0.783**	**0.735**	**0.812**	**0.825**	**0.762**	**0.771**
FCL(ACL+DCL)	**0.831**	**0.859**	**0.848**	**0.886**	**0.781**	**0.729**	**0.797**	**0.767**	**0.823**	**0.854**	**0.788**	**0.814**

TABLE IX

PERFORMANCE CHANGE WITH DIFFERENT TEST SET WHERE THE TRAINING SET INCLUDES T_1

training data (4.1–5.10)												
Method	T_2*(5.11–5.20)*		T_3*(5.21–5.31)*		T_4*(6.1–6.10)*		T_5*(6.11–6.20)*		T_6*(6.21–6.30)*		*Standard Deviation*	
	F_1	*AUC_PR*	F_1	*AUC_PR*	F_1	*AUC_PR*	F_1	*AUC_PR*	F_1	*AUC_PR*	F_1	*AUC_PR*
SL	0.802	0.807	0.657	0.531	0.683	0.597	0.749	0.717	0.711	0.647	0.0509	0.0955
LMSL	0.811	0.839	0.678	0.545	0.697	0.619	0.744	0.779	0.719	0.651	0.0461	0.1073
ASL	0.831	0.838	0.682	0.577	0.707	0.653	0.758	0.788	0.747	0.701	0.0509	0.0932
LMCL	0.822	0.851	0.702	0.641	0.717	0.692	0.781	0.813	0.730	0.699	0.0445	0.0792
AAML	0.834	0.847	0.729	0.684	0.764	0.708	0.797	0.826	0.742	0.733	0.0381	0.0650
ACL	**0.851**	**0.864**	**0.759**	**0.721**	**0.791**	**0.742**	**0.815**	**0.827**	**0.771**	**0.768**	**0.0328**	**0.0533**
FCL(ACL+DCL)	**0.853**	**0.877**	**0.797**	**0.725**	**0.799**	**0.771**	**0.820**	**0.859**	**0.789**	**0.823**	**0.0230**	**0.0562**

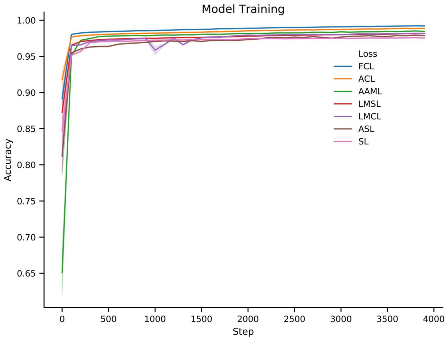


Fig. 4. Accuracy values of models in training process.

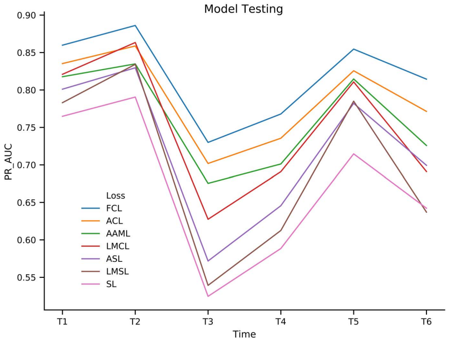


Fig. 6. Changes in PR_AUC values.

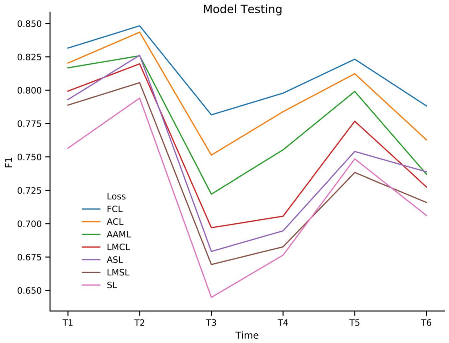


Fig. 5. Changes in F_1 values.

other one is used to demonstrate the performance stability of the deep representation learning model caused by our FCL. In all experiments, the best hyperparameters m, s, α, and γ for these loss functions are still searched by the grid search method.

In the first subexperiment, transactions from the first two months form the training set, and transactions from the last month form the testing set. This avoids the problem of data leakage. Fig. 4 shows the smoothed accuracy change of models when model training, and Table VII shows the results. Obviously, the deep representation learning models with our ACL and DCL converge faster, and our ACL and FCL still outperform other loss functions.

In the second subexperiment, transactions from the first month are used as the training set that is denoted as T, and the rest samples are used as the testing set. Instead of testing all samples of the testing set at one time, we divide them into six groups (from T_1 to T_6) in chronological order, and each group contains samples from ten consecutive days. We test these groups of data one by one to observe the performance changes. Table VIII shows the results, and it is obvious that ACL and FCL lead to the best performance in comparison with others. Then, just as shown by the curves in Figs. 5 and 6, the degrees of changes in our models' performance

TABLE X

PERFORMANCE CHANGE WITH DIFFERENT TEST SET WHERE THE TRAINING SET INCLUDES T_1 AND T_2

training data (4.1–5.20)										
Method	*T_3(5.21–5.31)*		*T_4(6.1–6.10)*		*T_5(6.11–6.20)*		*T_6(6.21–6.30)*		*Standard Deviation*	
	F_1	*AUC_PR*	*F_1*	*AUC_PR*	*F_1*	*AUC_PR*	*F_1*	*AUC_PR*	*F_1*	*AUC_PR*
SL	0.682	0.547	0.691	0.614	0.755	0.723	0.720	0.653	0.0285	0.0637
LMSL	0.688	0.553	0.699	0.629	0.753	0.786	0.728	0.657	0.0254	0.0840
ASL	0.691	0.583	0.715	0.667	0.768	0.791	0.751	0.702	0.0301	0.0745
LMCL	0.722	0.668	0.722	0.707	0.787	0.819	0.733	0.697	0.0269	0.0573
AAML	0.741	0.695	0.767	0.714	0.803	0.830	0.746	0.738	0.0244	0.0517
ACL	**0.770**	**0.732**	**0.797**	**0.749**	**0.819**	**0.833**	**0.779**	**0.772**	**0.0187**	**0.0382**
FCL(ACL+DCL)	**0.807**	**0.751**	**0.802**	**0.779**	**0.822**	**0.858**	**0.805**	**0.832**	**0.0077**	**0.0422**

are the lowest. This means that our ACL and FCL make the performance of models be the most stable.

To adequately show that our loss functions can lead to more stable performance, we construct two more data sets that includes training sets, $T + T_1$ and $T + T_1 + T_2$, and testing sets, $T_2 \sim T_6$ and $T_3 \sim T_6$, respectively. We compare our ACL and FCL with other methods on these two data sets, just like we did earlier. Tables IX and X show the results. Obviously, our loss functions lead to the least standard deviations, which means that our models ensure a more stable performance. At the same time, we can see that both F_1 and AUC_PR of our models are the best compared to others. From Tables VIII–X, it can be seen that when increasing the size of a training set, the performance of a trained model is enhanced. For example, from the results of T_3 in Tables VIII–X, the values of F_1 and AUC_PR of the corresponding models are all increased from Table VIII–X.

VI. CONCLUSION

In this article, a deep representation learning model is proposed for credit card fraud detection that has the advantage to achieve a good and stable performance. It consists of two parts: a deep neural network and a specially optimized loss function, FCL. FCL is able to supervise the deep representation learning model from both distance and angle so that the yielded model can enhance the intraclass compactness and interclass separation. Especially, ACL can directly separate learned representations of different classes in opposite directions. State-of-the-art loss functions are summarized and compared with our ACL and FCL. The experimental results illustrate the advantages of our method.

Although our loss functions can ensure more stable performance for fraud detection, there still is a space for improvement. For example, the performance stability of the fraud detection model should also be considered from the perspective of concept drift [5], [24]. In the future, we plan to consider the concept of drift problem from the aspect of loss function.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their constructive comments.

REFERENCES

[1] The Nilson Report. (2016). *Card Fraud Worldwide—The Nilson Report*. [Online]. Available: https://www.nilsonreport.com/upload/content_promo/The_ Nilson_ Report_ 10- 17-2016.pdf

[2] L. Zheng, G. Liu, C. Yan, and C. Jiang, "Transaction fraud detection based on total order relation and behavior diversity," *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 3, pp. 796–806, Sep. 2018.

[3] V. Van Vlasselaer *et al.*, "APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions," *Decis. Support Syst.*, vol. 75, pp. 38–48, Jul. 2015.

[4] K. Fu, D. Cheng, Y. Tu, and L. Zhang, "Credit card fraud detection using convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*. Cham, Switzerland: Springer, 2016, pp. 483–490.

[5] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Sep. 2018.

[6] J. Jurgovsky *et al.*, "Sequence classification for credit-card fraud detection," *Expert Syst. Appl.*, vol. 100, pp. 234–245, Jun. 2018.

[7] E. Kim *et al.*, "Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning," *Expert Syst. Appl.*, vol. 128, pp. 214–224, Aug. 2019.

[8] A. Dal Pozzolo, O. Caelen, Y.-A. Le Borgne, S. Waterschoot, and G. Bontempi, "Learned lessons in credit card fraud detection from a practitioner perspective," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4915–4928, Aug. 2014.

[9] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, no. 9, pp. 1263–1284, Jun. 2008.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jul. 2018.

[11] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.

[12] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[13] H. Yao, S. Zhang, R. Hong, Y. Zhang, C. Xu, and Q. Tian, "Deep representation learning with part loss for person re-identification," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2860–2871, Jun. 2019.

[14] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2884–2896, Nov. 2018.

[15] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 499–515.

[16] J. Dorronsoro, F. Ginel, C. Sgnchez, and C. Cruz, "Neural fraud detection in credit card operations," *IEEE Trans. Neural Netw.*, vol. 8, no. 4, pp. 827–834, Jul. 1997.

[17] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5718–5727, Apr. 2009.

[18] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[19] S. Nami and M. Shajari, "Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors," *Expert Syst. Appl.*, vol. 110, pp. 381–392, 2018.

[20] F. Zhang, G. Liu, Z. Li, C. Yan, and C. Jiang, "GMM-based undersampling and its application for credit card fraud detection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[21] R.-C. Chen, T.-S. Chen, and C.-C. Lin, "A new binary support vector system for increasing detection rate of credit card fraud," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 20, no. 02, pp. 227–239, Mar. 2006.

[22] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and C. Jiang, "Random forest for credit card fraud detection," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Mar. 2018, pp. 1–6.

[23] N. F. Ryman-Tubb, P. Krause, and W. Garn, "How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark," *Eng. Appl. Artif. Intell.*, vol. 76, pp. 130–157, Nov. 2018.

[24] D. Malekian and M. R. Hashemi, "An adaptive profile based fraud detection framework for handling concept drift," in *Proc. 10th Int. ISC Conf. Inf. Secur. Cryptol. (ISCISC)*, Aug. 2013, pp. 1–6.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[27] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Proc. Workshop Faces Real-Life Images, Detection, Alignment, Recognit.*, 2008, pp. 1–14.

[28] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2014, pp. 740–755.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[31] W. Brendel and M. Bethge, "Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet," 2019, *arXiv:1904.00760*. [Online]. Available: https://arxiv.org/abs/1904.00760

[32] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. Int. Conf. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 507–516.

[33] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 212–220.

[34] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 5265–5274.

[35] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.

[36] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2006, pp. 1735–1742.

[37] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2015, pp. 2892–2900.

[38] E. Ustinova and V. Lempitsky, "Learning deep embeddings with histogram loss," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4170–4178.

[39] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*. [Online]. Available: https://arxiv.org/abs/1703.07737

[40] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-center loss for multi-view 3D object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1945–1954.

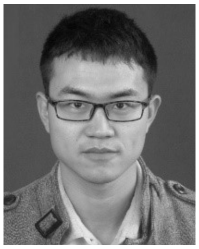
[41] A. Correa Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Feature engineering strategies for credit card fraud detection," *Expert Syst. Appl.*, vol. 51, pp. 134–142, Jun. 2016.

[42] S. Wang, G. Liu, Z. Li, S. Xuan, C. Yan, and C. Jiang, "Credit card fraud detection using capsule network," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 3679–3684.

[43] Machine Learning Group. (2018). *Credit Card Fraud Detection*. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud

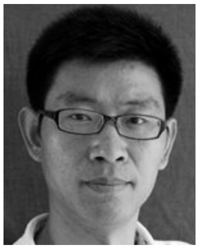
[44] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in *Proc. Eur. Conf. Inf. Retr. (ECIR)*. Berlin, Germany: Springer, 2005, pp. 345–359.

[45] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, Mar. 2015, Art. no. e0118432.



Zhenchuan Li received the bachelor's degree in automation from Shanghai University, Shanghai, China, in 2015. He is currently pursuing the Ph.D. degree in computer science and technology with Tongji University, Shanghai.

His research interests include machine learning, deep learning, and fraud detection with several published articles in some journals and conferences in these areas.



Guanjun Liu (Senior Member, IEEE) received the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2011.

He was a Post-Doctoral Research Fellow with the Singapore University of Technology and Design, Singapore, from 2011 to 2013, and a Post-Doctoral Research Fellow with the Humboldt University of Berlin, Berlin, Germany, from 2013 to 2014, supported by the Alexander von Humboldt Foundation. He is currently a Professor with the Department of Computer Science, Tongji University.

Changjun Jiang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995.

He is currently the Leader of the Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Tongji University, Shanghai, China. He is also an Honorary Professor with Brunel University London, Uxbridge, England.

Dr. Jiang is also an IET Fellow. He has been the recipient of one international prize and seven prizes in the field of science and technology.