Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)
Special Track on AI in FinTech

Federated Meta-Learning for Fraudulent Credit Card Detection

Wenbo Zheng [1,2], **Lan Yan** [2,4], **Chao Gou** [3 *] and **Fei-Yue Wang** [2,4]
[1] School of Software Engineering, Xi'an Jiaotong University
[2] The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences
[3] School of Intelligent Systems Engineering, Sun Yat-sen University
[4] School of Artificial Intelligence, University of Chinese Academy of Sciences
zwb2017@stu.xjtu.edu.cn, yanlan2017@ia.ac.cn, gouchao@mail.sysu.edu.cn, feiyue.wang@ia.ac.cn

Abstract

Credit card transaction fraud costs billions of dollars to card issuers every year. Besides, the credit card transaction dataset is very skewed, there are much fewer samples of frauds than legitimate transactions. Due to the data security and privacy, different banks are usually not allowed to share their transaction datasets. These problems make traditional model difficult to learn the patterns of frauds and also difficult to detect them. In this paper, we introduce a novel framework termed as **Federated Meta-Learning** for fraud detection. Different from the traditional technologies trained with data centralized in the cloud, our model enables banks to learn fraud detection model with the training data distributed on their own local database. A shared whole model is constructed by aggregating locally-computed updates of fraud detection model. Banks can collectively reap the benefits of shared model without sharing the dataset and protect the sensitive information of cardholders. To achieve the good performance of classification, we further formulate an improved triplet-like metric learning, and design a novel meta-learning-based classifier, which allows joint comparison with K negative samples in each mini-batch. Experimental results demonstrate that the proposed approach achieves significantly higher performance compared with the other state-of-the-art approaches.

1 Introduction

In recent years, with the popularity of credit cards and the rapid development of electronic services such as e-commerce, e-finance, and mobile payments, the amount of credit card transactions has increased dramatically. The large-scale use of credit cards and various transaction scenarios without strict verification and supervision will inevitably lead to billions of dollars in losses due to credit card fraud. It is difficult to obtain an accurate estimate of the loss since card issuers are often reluctant to release the statistics. However, there is some publicly available data that show the severity of credit card fraud. According to the Nilson Report [Carneiro *et al.*, 2017], the losses due to credit and debit card fraud reached $16.31 billion in 2014. Cybersource reported that around online fraud caused a $3.5 billion dollar loss in 2012 [Mahmoudi and Duman, 2015].

Credit card fraud detection is an important way to prevent fraud events, which is usually categorized into two techniques: 1) anomaly detection and 2) classifier-based detection. Anomaly detection focuses on calculating the distance among the data points in space. By calculating the distance between the incoming transaction and the cardholder's profile, an anomaly detection method can filter any incoming transaction which is inconsistent with the cardholder's profile. The second technique utilizes machine learning methods to train a classifier through the use of supervised binary classification systems properly trained from pre-screened sample datasets [Bahnsen *et al.*, 2014]. Recently, deep-learning-based methods [Ki and Yoon, 2018] present a promising solution to the problem of credit card fraud detection by enabling institutions to make optimal use of their historic customer data as well as real-time transaction details that are recorded at the time of the transaction. Deep-learning-based provide comparable results to prevailing fraud detection methods. However, in such a specific application domain, datasets available for training are strongly imbalanced, with the class of interest considerably less represented than the other. This significantly reduces the effectiveness of binary classifiers, undesirably biasing the results toward the prevailing class, while we are interested in the minority class. Oversampling the minority class has been adopted to alleviate this problem, but this method still has some drawbacks, one of which is that does not add informative content, thus limiting the improvement on the ability of a classifier to generalize.

In general, there are two main challenges in the task of fraudulent credit card detection:

(1) *The public available dataset is imbalanced and limited.* It is intrinsic that the imbalance lies in the natural data space, and about 2% of the entire credit card transactions constitute as fraud activities. In real world, due to the data security and privacy, different banks are usually not allowed to share their transaction datasets.

(2) *Exploring the relationship among training samples is neglected.* The traditional technique only focuses on the comparison with samples from different classes.

*Chao Gou is the corresponding author.

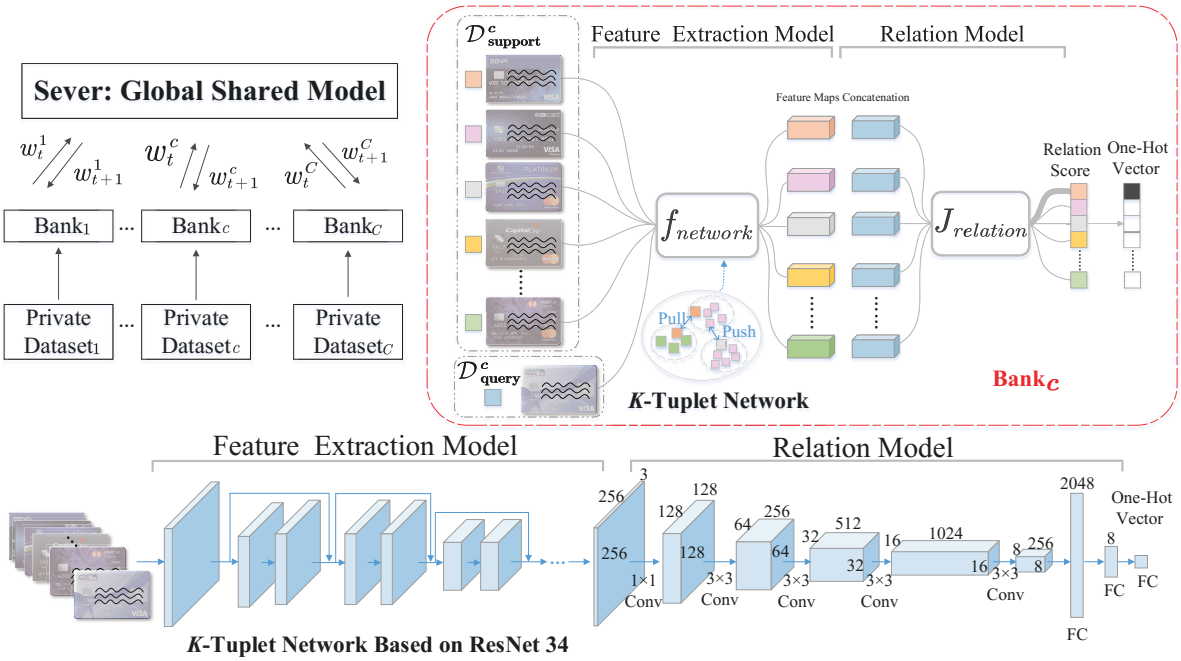


Figure 1: The Pipeline of Federated Meta-Learning Based Approach. There are fixed set of C banks as participants in the whole framework. $w_t^c(c = 1, 2, \cdots, C)$ represents the banks parameter that upload to server, w_{t+1}^c represents the parameter that averaging by server. Meta-classifier in each back c, contains two modules: a feature extraction model and a relation model. The feature extraction model $f_{network}$ produces feature maps to represent feature extraction function. The relation model $J_{relation}$ represents the similarity between support set $\mathcal{D}^c{}_{\text{support}}$ and query set $\mathcal{D}^c{}_{\text{query}}$. The classifier uses designed K-Tuplet Network based on ResNet-34 as feature extraction model. The output of this network can be regarded as network features, and then we apply the relation model.

In contrast, even though there is a few samples are fraud while a majority of them are legitimate transactions, humans are very good at recognizing the subjects. Why can human beings recognize it quickly and accurately with very little direct supervision, or none at all? Probably because human beings can use the our experience to learn. And isn't this one of the mechanisms of meta-learning [Zheng *et al.*, 2019a; Zheng *et al.*, 2019b; Zheng *et al.*, 2020b; Zheng *et al.*, 2020c]? *So how we use the principle of meta-learning to build models for fraudulent credit card detection?*

Inspired by curriculum learning and negative exemplar mining strategy, networks based on triplet loss [Schroff *et al.*, 2015] are proposed to learn features, and get the goal that samples from the same class should be close together. Therefore, *how we design a novel model using this strategy to ensure the efficiency to handle unseen class samples?*

Furthermore, federated learning is an emerging artificial intelligence basic technology [Yang *et al.*, 2019]. The goals of federated learning are to ensure information security during big data exchange, and protect terminal data and personal data privacy. Thus, *why not incorporate federated learning techniques into our model?*

To tackle all the aforementioned problems, in this paper, we propose a novel meta-learning-based model for the fraudulent credit card detection task, which exploits federated learning strategy in the whole process. Specially, we formulate an improved triplet-like metric learning, namely the *deep K-tuplet network*, to achieve few-shot classification. Particularly, this network generalizes the triplet network to allow joint comparison with K negative samples in each mini-batch. Further, we design a federated-learning-based model based on our K-tuplet network, which can protect the data privacy, meanwhile, it can be shared with different banks. This framework also enables different banks to collaboratively learn a shared model while keeping all the training data which is skewed on their own private database. We experimentally demonstrate that the proposed approach achieves significantly higher performance compared with the state-of-the-art approaches, using four open-source datasets. Besides, qualitative discussion demonstrates that meta-learning-based proposed strategy achieves significantly higher performance compared with the meta-learning-based others.

In summary, our main contributions are as follows:

☼ We propose a novel fraudulent credit card detection approach. To the best of our knowledge, this is the first attempt to study the fraudulent credit card detection approach based on federated meta-learning.

☼ In order to adaptively integrate with the meta-based model, we design a novel federated learning framework, which not only guarantees data privacy but also has been experimentally proven to be lossless.

☼ We present a novel meta-learning-based classifier to effectively learn the discriminative feature extraction on unseen class samples. Qualitative discussion demonstrates that this

strategy achieves competitive performance over meta-based others for fraudulent credit card detection task.

☼ Our network is effective, and experimental results show that the proposed approach has strong robustness and outperforms existing methods.

2 Federated Meta-Learning Based Approach

In this section, we present our approach and the pipeline is shown in Figure 1.

2.1 Problem Setup

We consider the problem of fraudulent credit card detection as meta-learning- and federated-learning-based (federated meta-learning based) classification.

In federated fraud detection framework, let D_i denotes a credit card transaction dataset, (x_i, y_i) is the training data sample of D_i with a unique index i. Vector $x_i \in R^d$ is a d-dimensional real-valued feature vector, which is regarded as the input of the fraud detection model. Scalar $y_i \in \{0, 1\}$ is a binary class label, which is the desired output of the model. $y_i = 1$ denotes that it is a fraud transaction, $y_i = 0$ denotes that it is a normal transaction. To facilitate the learning process, every model has a loss function defined on its parameter vector w for each data sample. The loss function captures the error of the fraud detection model on the training data. The loss of the prediction on a sample (x_i, y_i) made with the model parameters w, we define it as $l(x_i; y_i; w)$. The learning rate controls the speed that model converges to the best accuracy. We define the learning rate as η.

Suppose there are fixed set of C banks (or financial institutions) as participants, each bank possesses a fixed private dataset $D_i = \{{x_i}^c, {y_i}^c\}$ $(c = 1, 2, 3, \ldots, C)$. ${x_i}^c$ is the credit card transaction sample, ${y_i}^c$ is the corresponding label and n_c is the size of dataset associated with participant bank c.

For each participant bank c, we use meta learning, which consists of two phases: meta-training and meta-testing. In meta-training, our training data $\mathcal{D}^c{}_{\text{meta - train}} = \{({x_i}^c, {y_i}^c)\}_{i=1}^{n_c^{train}}$ from a set of classes $\mathcal{C}_{train}$ are used for training a classifier, where ${x_i}^c$ is the credit card transaction training sample, ${y_i}^c \in \mathcal{C}_{train}$ is the corresponding label, and n_c^{train} is the number of training samples. In meta-testing, a support set of n_c^{support} labeled examples $\mathcal{D}^c{}_{\text{support}} = \{({x_j}^c, {y_j}^c)\}_{j=1}^{n_c^{\text{support}}}$ from a set of classes $\mathcal{C}_{test}$ is given, where ${x_j}^c$ is a credit card transaction test sample, and ${y_i}^c \in \mathcal{C}_{test}$ is the corresponding label. The goal is to predict the labels of a query set $\mathcal{D}^c{}_{\text{query}} = \{(x_j)\}_{j=n_c^{\text{support}}+1}^{n_c^{\text{support}}+n_c^{\text{query}}}$, where n_c^{query} ($n_c^{\text{query}} = n_c - n_c^{train} - n_c^{\text{support}}$) is the number of queries. Further, we use the meta-learning on the training set to transfer the extracted knowledge to on the support set. It aims to perform the network's learning on the support set better and classify the query set more successfully. The process is described in Algorithm 1. From Algorithm 1 and Figure 1, our federated meta-learning can be concluded to five steps:

Step 1: Participating bank downloads the common meta-learning based classifier from the server, which plays role as a global shared model;
Step 2: Improving the classifier by learning data on each bank;
Step 3: Summarizes the changes of the classifier as a small focused update and send it using encrypted communication to the server;
Step 4: The server immediately aggregates with all banks updates to improve each classifier;
Step 5: The process repeats until convergence.

Algorithm 1: Federated Meta-Learning Approach

Input: The private dataset of banks
Output: Our credit card fraud detection model

Procedure `Server-Update`
  Initialize the detection classifier and its parameters w_0;
  for *each round* $t = 1, 2, 3, \ldots, T$ **do**
    /* *seed* is the fraction of banks that be selected to perform computation on each round */
    Random choose $max(seed \times C, 1)$ banks as N_t;
    for *each banks* $c \in N_t$ *in parallel* **do**
      /* $\left\|L^c_{relation_{t+1}}\right\|$ is the value of $L^c_{relation}$ at the $t+1$ round */
      $w^c_{t+1}, \left\|L^c_{relation_{t+1}}\right\| \leftarrow$ `Bank-Update`(n_{whole}, w_t, c);
      $w_{t+1} \leftarrow \sum_{t=1}^{T} \frac{n_c}{n_{whole}} \times \left\|L^c_{relation_{t+1}}\right\| \times w^c_{t+1}$;
Procedure `Bank-Update`(n_{whole}, w, c)
  Sample support set $\mathcal{D}^c{}_{\text{support}}$ and training set $\mathcal{D}^c{}_{\text{meta - train}}$;
  `Training`(**Meta-Learning Based Classifier**);
  $w^c_{t+1} \leftarrow w^c_t - \eta \times \nabla L^c_{relation}({x_i}^c; {y_i}^c; w)$;
  `Testing`(**Meta-Learning Based Classifier**);
  return w^c_{t+1} and $\left\|L^c_{relation_{t+1}}\right\|$ to server

2.2 Meta-Learning for Fraudulent Detection

Considering that (1) nonlinear mapping should be generalizable to work with samples of novel classes, and (2) the mapping should preserve the class relationship on the unseen class samples in $\mathcal{D}^c{}_{\text{support}}$ and $\mathcal{D}^c{}_{\text{query}}$, we propose a novel matching networks based on relational network [Sung *et al.*, 2018] to solve the problem of fraudulent credit card detection. First, we meta-learn a transferable feature exaction model through *the deep K-tuplet network* with the designed K-tuplet loss from the training dataset. The well-learned features of the query samples in the support set are then fed into the nonlinear distance metric to learn the similarity scores. Further, we conduct few-shot classification based on these scores. As illustrated in Figure 1, our matching network consists of two branches: a **feature extraction model** and a **relation model** during the training of our network.

Meta-Learning Based Feature Extraction The training samples from $\mathcal{D}^c{}_{\text{meta - train}}$ are randomly selected to form a triplet $({x^c}_a; {x^c}_p; {x^c}_n)$ with an anchor sample ${x^c}_a$, a positive

sample ${x^c}_p$, and a negative sample ${x^c}_n$. The label of the selected samples in a triplet should satisfy ${y^c}_a = {y^c}_p \neq {y^c}_n$. The aim of our feature extraction is to pull the feature maps of anchor and positive samples close to each other, while pushing the feature maps of anchor and negative samples far apart.

In particular, we randomly choose the K negative samples ${x^c}_{n_i} (i = 1, 2, 3, \ldots, K)$ to form into a triplet. We define the function $f_{network}$ which represents feature extraction function using network to produce feature maps $f_{network}({x^c}_a)$, $f_{network}({x^c}_p)$ and $f_{network}({x^c}_n)$. We design K-tuplet loss during the training procedure of feature extraction:

$$\begin{aligned} L^{feature}_{extraction}({x^c}_a; {x^c}_p; {x^c}_{n_i}) = \\ \frac{1}{K} \sum_{i=1}^{K} [\|f_{network}({x^c}_a) - f_{network}({x^c}_p)\|^2 \\ - \|f_{network}({x^c}_a) - f_{network}({x^c}_{n_i})\|^2 \\ + \alpha_{train}]_+ \end{aligned} \tag{1}$$

where $[\cdot]_+ = \max(0, \cdot)$ denotes the hinge loss function and α_{train} is the hyper-parameter margin. For the anchor sample ${x^c}_a$, the optimization shall maximize the distance to the negative samples ${x^c}_{n_i}$ to be larger than the distance to the positive sample ${x^c}_p$ in the feature space. To form one mini-batch to train the network, we randomly select B anchor samples from the training set, where B is batch size. For each anchor sample ${x^c}_a$, we then randomly select another positive sample ${x^c}_p$ of the same class as ${x^c}_a$ and further randomly select K other negative samples whose classes are different from ${x^c}_a$. Among the K negative samples, their class labels may be different. Compared with the traditional triplet loss, each forward update in our K-tuplet loss considers more inter-class variations, thus making the learned feature embedding more discriminative for samples from different classes.

Meta-Learning Based Relation Model We further nonlinear distance relation model to learn to compare the sample features in few-shot classification.

Given sample ${x_j}^c$ in support set ${\mathcal{D}^c}_{\text{support}}$ and sample ${x_i}^c$ in the train set ${\mathcal{D}^c}_{\text{meta - train}}$, we assume the $C_{network}(\cdot, \cdot)$ to be concatenation of corresponding feature maps in depth. The combined feature map of the sample and query is used as the relation model $J_{relation}(\cdot)$ to get a scalar in range of 0 to 1 representing the similarity between ${x_i}^c$ and ${x_j}^c$, which is called relation score. Suppose we have one labeled sample for each of n_c^{train} unique classes, our model can generate n_c^{train} relation scores $Judge_{i,j}$ for the relation between one query input ${x_j}^c$ and training sample set examples ${x_i}^c$:

$$\begin{aligned} &Judge_{i,j} = J_{relation}(C_{network}(f_{network}({x_j}^c), f_{network}({x_i}^c))) \\ &i = 1, 2, \cdots, n_c^{train} \end{aligned} \tag{2}$$

Furthermore, we can do the operation of element-wise sum over our feature extraction model outputs of all samples from each training class to form this class's feature map. And this pooled class-level feature map is concatenated with the feature map of the test samples as above.

We use mean square error (MSE) loss to train our relation model, regressing the relation score $Judge_{i,j}$ to the ground truth: matched pairs have similarity 1 and the mismatched pair have similarity 0.

$$L^c_{relation} = \arg\min \sum_{i=1}^{n_c^{train}} \sum_{j=1}^{n_c^{\text{support}}} (Judge_{i,j} - ((y_i{}^c == y_j{}^c)^2 \tag{3}$$

Network Architecture Our network architecture is shown in Figure 1. Our network consists of two parts.

We employ the ResNet-34 architecture [He *et al.*, 2016] for learning the feature exaction model. When meta-learn the transferable feature exaction, we use Adam optimizer [Kingma and Ba, 2014] with a learning rate of 0.001 and a decay for every 40 epochs. We totally train 1000 epochs and adopt the semi-hard mining strategy [Harwood *et al.*, 2017] when the loss starts to converge.

We use the 8-layer network architecture. Taking an sample feature map as input, the output of the 8-th pooling layer is one-hot vector. The kernels of network change in turns: $3 \times 256 \times 256 \rightarrow 128 \times 128 \times 128$ (Convolution, kernel size: 1×1) $\rightarrow 256 \times 64 \times 64$ (Convolution, kernel size: 3×3) $\rightarrow 512 \times 32 \times 32$ (Convolution, kernel size: 3×3) $\rightarrow 1024 \times 16 \times 16$ (Convolution, kernel size: 3×3) $\rightarrow 256 \times 8 \times 8$. Then, we apply the fully connected layer to change into 2048-dimensional vector. Finally, We use two fully-connected layers to have 8 and 1 outputs respectively, followed by a sigmoid function to get the final similarity scores mention in Eq. (3). Other network settings are similar to our feature extraction model.

2.3 The Federated Meta-Learning Framework

In our fraud detection system, we use federated learning to get the goal that different banks can share dataset to build an effective fraud detection model without revealing the privacy of each bank's customers. Before getting involved in training the fraud detection model, all banks will first agree on a common fraud detection model (the architecture of the model, activation function in each hidden layer, loss function, etc). *Obviously, we choose the horizontal federated learning framework*, which is introduced in the scenarios that data sets share the same feature space but different in samples.

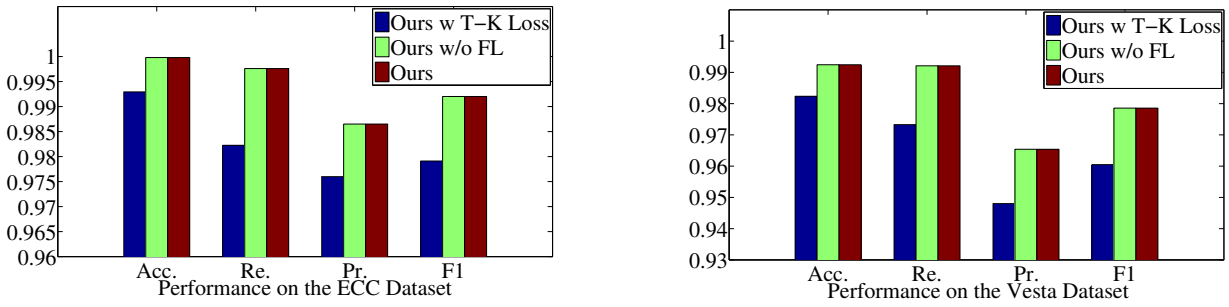
In federated fraud detection model, we use $n_{whole} = \sum_{c=1}^{C} n_c$ to represent all the data samples involved in the whole process. We can get the objective loss function:

$$\begin{aligned} &\min \mathfrak{l}(x_i; y_i; w), \\ &\text{where } \mathfrak{l}(x_i; y_i; w) = \frac{1}{n_c} \sum_{i \in D_i} L^c_{relation}({x_i}^c; {y_i}^c; w) \end{aligned} \tag{4}$$

The server will initialize the fraud detection model parameters. At each communication round t ($t = 1, 2, 3, \ldots, T$), and a random fraction $seed$ of banks will be selected. These banks will communicate directly with the server. First, download the current global model parameters from the server. Then, every bank computers the average gradient of the loss f_c on their own private dataset at current fraud detection model parameters with a fixed learning rate η, $f_c =$

Dataset	ECC				RA				SD				Vesta			
Performance	Acc.	Re.	Pr.	F1	Acc.	Re.	Pr.	F1	Acc.	Re.	Pr.	F1	Acc.	Re.	Pr.	F1
BMR	69.26%	71.12%	66.83%	68.91%	22.26%	37.13%	50.13%	42.66%	49.12%	8.26%	28.07%	12.77%	30.56%	15.73%	40.03%	22.59%
APATE	70.22%	71.57%	66.84%	69.12%	28.45%	56.80%	59.63%	58.18%	63.96%	16.55%	28.27%	20.88%	37.92%	24.33%	55.95%	33.92%
PD-FDS	70.95%	74.00%	68.39%	71.08%	29.53%	60.91%	60.15%	60.53%	66.36%	28.66%	48.50%	36.02%	52.78%	30.47%	60.25%	40.47%
SPD	74.51%	82.15%	69.27%	75.16%	55.64%	66.03%	71.86%	68.82%	67.65%	74.85%	48.78%	59.07%	58.58%	32.55%	70.84%	44.60%
CMAB	81.14%	86.14%	71.45%	78.11%	79.20%	66.43%	82.33%	73.53%	76.45%	76.07%	48.96%	59.58%	75.39%	60.72%	78.05%	68.30%
RawLR	81.18%	91.81%	82.42%	86.86%	79.70%	72.92%	82.88%	77.58%	76.76%	80.45%	49.16%	61.03%	85.90%	73.53%	83.29%	78.10%
RMNLS	83.81%	92.49%	89.08%	90.75%	87.78%	76.55%	83.41%	79.83%	82.25%	81.39%	74.85%	77.98%	88.29%	78.16%	85.18%	81.52%
FlowScope	87.86%	93.05%	89.32%	91.15%	89.03%	98.92%	84.88%	91.36%	95.82%	83.32%	88.47%	85.82%	89.95%	90.41%	97.37%	93.76%
Ours	99.98%	99.76%	98.65%	99.20%	99.91%	99.01%	98.06%	98.53%	99.02%	99.01%	97.68%	98.34%	99.25%	99.21%	96.54%	97.86%

Table 1: Performance Comparison with State-of-the-Art Baselines



(a) Performance Comparison on the ECC Dataset

(b) Performance Comparison on the Vesta Dataset

Figure 2: Results of The Ablation Experiment

$\nabla L^c_{relation}(x_i{}^c; y_i{}^c; w)$. These banks update their fraud detection model synchronously and send the update of fraud detection model to server. The details of our fraud detection model training process are described in Algorithm 1.

2.4 Comparison with Federated Learning

Conceptually, federated learning provides an approach to share data information at the model level. Although trained in a distributed manner, the unified model needs to take all data input into consideration and to provide fraud detection for all banks, which suffers from the large size necessary in many practical circumstances. Federated meta-learning, on the other hand, provides an approach to share data information at the higher meta-learner level, making it possible to train small task-specific or data-specific models. Technically, in federated learning the transmission between the server and bank devices involves current models, while in federated meta-learning the transmission involves the meta-learner from server and test feedback from banks to improve the meta-learner's training capacity.

3 Experiments and Results

We evaluated the performance of our algorithm by accuracy (Acc.), recall (Re.), precision (Pr.) and F-measure (F1).

3.1 Dataset Description

We conducted multiple experiments on four datasets.

- **ECC**: We sourced the first dataset from the European Credit Card (ECC) transactions provided by the ULB ML Group [Dal Pozzolo, 2015]. This dataset contains anonymized $284,807$ highly imbalanced credit card transactions with an Imbalance Ratio of $1:578$, and 0.17% or 492 of fraudulent transactions.
- **RA**: We sourced the second dataset from the Revolution Analytics (RA)[Mohammed *et al.*, 2018]; the dataset contains 10 million credit card transactions, which is massive with imbalanced ratio of $1:16$, and consists of 5.96% of fraudulent transactions.
- **SD** and **Vesta**: We sourced the third and fourth datasets from Kaggle; the third dataset is synthetic dataset (SD) [1] to evaluate the performance of fraud detection methods; the fourth dataset [2], is a challenging large-scale dataset, which comes from Vesta's real-world e-commerce transactions and contains a wide range of features from device type to product features.

3.2 Comparison Experiment

In this subsection, we compare the state-of-the-art baselines with our model on four datasets.

Training In this paper, we set the 8 banks to join this mechanism for all experiments, that is, we set C to 8.

Baselines We compare against various state-of-the-art baselines, including BMR [Bahnsen *et al.*, 2014], APATE [Lebichot *et al.*, 2017], PD-FDS [Ki and Yoon, 2018], SPD [Braun *et al.*, 2017], CMAB [Soemers *et al.*, 2018], RawLR [Jiang *et al.*, 2018], RMNLS [Dal Pozzolo *et al.*, 2018], and FlowScope [Xiangfeng Li, 2020].

[1] https://www.kaggle.com/ntnu-testimon/paysim1

[2] https://www.kaggle.com/c/ieee-fraud-detection/overview

Comparison with State-of-the-Art Methods From Table 1, our model is better than others on four datasets. It can get the following two points:

Firstly, baselines except for FlowScope are deep-learning-based approaches. Since these approaches cannot effectively handle imbalanced data sets, the performances of these approaches is lower than ours.

Secondly, FlowScope is the graph-learning-based approach. As the graph learning method mainly deals with data with a lot of relationships, not data with random relationships, the performance of FlowScope is lower than ours.

From the above two points, *it is clear that the design of federated meta-learning is more effective than deep learning or graph learning for fraudulent credit card detection.*

3.3 Ablation Experiment

In this subsection, in order to verify the reasonableness and effectiveness of *K-tuplet loss* and *Federated Learning*, we design the ablation experiment. In Figure 2, "**Ours w/o FL**" means a variant of Ours, which removes federated learning strategy, and at this point, we use traditional meta-learning strategy; "**Ours w T-K Loss**" means a variant of Ours, which removes K-tuplet loss and use traditional K-Triplet loss [Schroff *et al.*, 2015]. We analyze the following two aspects:

Ablation study about the K-Tuplet Loss Compared to "Ours w T-K Loss", it is obvious that the results of ours is better than "Ours w T-K Loss" from Figure 2(a). This means, once trained, our network is able to extract discriminative features for unseen novel categories and can be seamlessly incorporated with a non-linear distance metric function to facilitate the few-shot classification. This also suggests our design of K-Tuplet loss is able to help us to improve fraudulent credit card detection.

Ablation study about Federated Learning Compared to "Ours w/o FL", it is obvious that the results of ours is same as "Ours w/o FL" from Figure 2(a). This means though both servers and banks are done data protection, our approach can be performed without loss of performance.

Moreover, by analyzing ablation results shown in Figure 2(b), on Vesta dataset, we can get similar conclusions.

3.4 Discussion about Different Meta-Learning

In subsection, we compare with state-of-the-art meta-learning approaches, to verify the effectiveness of ours.

We can divide meta-learning methods into three categories [Sun *et al.*, 2019]:

(1) Metric learning methods (i.e., MatchingNets, RelationNets, and ours) learn a similarity space in which learning is particularly efficient for few-shot training examples.

(2) Memory network methods (i.e., Meta Networks, TADAM) learn to store "experience" when learning seen tasks and then generalize it to unseen tasks.

(3) Gradient descent based meta-learning methods (i.e., MAML, LEO, LGM-Net, CTM) intend for adjusting the optimization algorithm so that the model can converge within a small number of optimization steps (with a few examples).

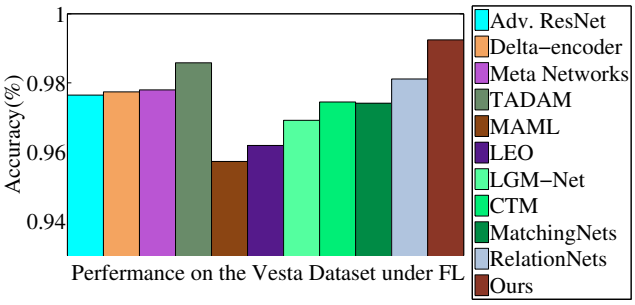


Figure 3: Performance Comparison on The Discussion Experiment

From Figure 3, ours is better than others. This means our design of meta-learning is better than other meta-learning approaches for fraudulent credit card detection. Besides, the metric learning methods are better than memory network methods or gradient descent based meta-learning methods. *Through the above analysis, we can find that our meta-learning achieves the best performance among the state-of-the-art other meta-learning approaches.*

4 Conclusion and Future Work

In this paper, we propose an novel and effective federated meta-learning based fraudulent credit card detection model. An novel meta-learning-based classifier, namely the deep K-tuplet network, is designed. This network generalizes the triplet network to allow joint comparison with K negative samples in each mini-batch. Furthermore, different from the traditional technologies trained with data centralized in the cloud, our model based on federated learning, enables banks to learn fraud detection model with the training data distributed on their own local database. A shared whole model is constructed by aggregating locally-computed updates of fraud detection model. Experimental results demonstrate that the proposed approach achieves significantly higher performance compared with the state-of-the-art approaches.

In future research, on the one hand, we consider to investigate resource consumption of the federated meta-learning framework. On the other hand, we plan to study how to implement an algorithm in the blockchain [Zheng *et al.*, 2020a].

Acknowledgements

We would like to thank the anonymous reviewers for their useful feedback. This work is supported in part by the Key Research and Development Program 2020 of Guangzhou, MOST and NNSF of China (2008AAA0101502, 61533019, 61806198, U1811463), and Squirrel AI Learning.

References

[Bahnsen *et al.*, 2014] Alejandro Correa Bahnsen, Aleksandar Stojanovic, Djamila Aouada, and Björn Ottersten. Improving credit card fraud detection with calibrated probabilities. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 677–685, 2014.

[Braun *et al.*, 2017] Fabian Braun, Olivier Caelen, Evgueni N. Smirnov, Steven Kelk, and Bertrand Lebichot. Improving card fraud detection through suspicious pattern discovery. In Salem Benferhat, Karim Tabia, and Moonis Ali, editors, *Advances in Artificial Intelligence: From Theory to Practice*, pages 181–190, Cham, 2017. Springer International Publishing.

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

[Dal Pozzolo *et al.*, 2018] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi. Credit card fraud detection: A realistic modeling and a novel learning strategy. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3784–3797, Aug 2018.

[Dal Pozzolo, 2015] Andrea Dal Pozzolo. Adaptive machine learning for credit card fraud detection. 2015.

[Harwood *et al.*, 2017] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Jiang *et al.*, 2018] C. Jiang, J. Song, G. Liu, L. Zheng, and W. Luan. Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism. *IEEE Internet of Things Journal*, 5(5):3637–3647, Oct 2018.

[Ki and Yoon, 2018] Youngjoon Ki and Ji Won Yoon. Pdfds: Purchase density based online credit card fraud detection system. In Archana Anandakrishnan, Senthil Kumar, Alexander Statnikov, Tanveer Faruquie, and Di Xu, editors, *Proceedings of the KDD 2017: Workshop on Anomaly Detection in Finance*, volume 71 of *Proceedings of Machine Learning Research*, pages 76–84. PMLR, 14 Aug 2018.

[Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec 2014.

[Lebichot *et al.*, 2017] Bertrand Lebichot, Fabian Braun, Olivier Caelen, and Marco Saerens. A graph-based, semi-supervised, credit card fraud detection system. In Hocine Cherifi, Sabrina Gaito, Walter Quattrociocchi, and Alessandra Sala, editors, *Complex Networks & Their Applications V*, pages 721–733, Cham, 2017. Springer International Publishing.

[Mahmoudi and Duman, 2015] Nader Mahmoudi and Ekrem Duman. Detecting credit card fraud by modified fisher discriminant analysis. *Expert Systems with Applications*, 42(5):2510 – 2516, 2015.

[Mohammed *et al.*, 2018] Rafiq Ahmed Mohammed, Kok-Wai Wong, Mohd Fairuz Shiratuddin, and Xuequn Wang. Scalable machine learning techniques for highly imbalanced credit card fraud detection: A comparative study. In Xin Geng and Byeong-Ho Kang, editors, *PRICAI 2018: Trends in Artificial Intelligence*, pages 237–246, Cham, 2018. Springer International Publishing.

[Schroff *et al.*, 2015] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, June 2015.

[Soemers *et al.*, 2018] Dennis JNJ Soemers, Tim Brys, Kurt Driessens, Mark HM Winands, and Ann Nowé. Adapting to concept drift in credit card transaction data streams using contextual bandits and decision trees. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Sun *et al.*, 2019] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[Sung *et al.*, 2018] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, June 2018.

[Xiangfeng Li, 2020] Zifeng Li Xiaotian Han Chuan Shi Bryan Hooi He Huang Xueqi Cheng Xiangfeng Li, Shenghua Liu. Flowscope: Spotting money laundering based on graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2020.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):12:1–12:19, January 2019.

[Zheng *et al.*, 2019a] W. Zheng, L. Yan, C. Gou, W. Zhang, and F. Wang. A relation network embedded with prior features for few-shot caricature recognition. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1510–1515, July 2019.

[Zheng *et al.*, 2019b] Wenbo Zheng, Chao Gou, Lan Yan, and Wang Fei-Yue. A relation hashing network embedded with prior features for skin lesion classification. In *Machine Learning in Medical Imaging*, pages 115–123, Cham, 2019.

[Zheng *et al.*, 2020a] W. Zheng, K. Wang, and F. Wang. Gan-based key secret-sharing scheme in blockchain. *IEEE Transactions on Cybernetics*, pages 1–12, 2020.

[Zheng *et al.*, 2020b] Wenbo Zheng, Chao Gou, and Fei-Yue Wang. A novel approach inspired by optic nerve characteristics for few-shot occluded face recognition. *Neurocomputing*, 376:25 – 41, 2020.

[Zheng *et al.*, 2020c] Wenbo Zheng, Chao Gou, Lan Yan, and Shaocong Mo. Learning to classify: A flow-based relation network for encrypted traffic classification. In *Proceedings of The Web Conference 2020*, WWW '20, page 13–22, New York, NY, USA, 2020.