FFD: A Federated Learning Based Method for Credit Card Fraud Detection

Wensi Yang[1,2], Yuhang Zhang[1,2], Kejiang Ye[1], Li Li[1],
and Cheng-Zhong Xu[3](✉)

[1] Shengzhen Institutes of Advanced Technology, Chinese Academy of Sciences,
Shenzhen 518055, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Department of Computer and Information Science,
Faculty of Science and Technology, State Key Laboratory of IoT for Smart City,
University of Macau, Taipa, Macao, Special Administrative Region of China
cz.xu@siat.ac.cn

Abstract. Credit card fraud has caused a huge loss to both banks and consumers in recent years. Thus, an effective Fraud Detection System (FDS) is important to minimize the loss for banks and cardholders. Based on our analysis, the credit card transaction dataset is very skewed, there are much fewer samples of frauds than legitimate transactions. Furthermore, due to the data security and privacy, different banks are usually not allowed to share their transaction datasets. These problems make FDS difficult to learn the patterns of frauds and also difficult to detect them. In this paper, we propose a framework to train a fraud detection model using behavior features with federated learning, we term this detection framework FFD (Federated learning for Fraud Detection). Different from the traditional FDS trained with data centralized in the cloud, FFD enables banks to learn fraud detection model with the training data distributed on their own local database. Then, a shared FDS 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. Furthermore, an oversampling approach is combined to balance the skewed dataset. We evaluate the performance of our credit card FDS with FFD framework on a large scale dataset of real-world credit card transactions. Experimental results show that the federated learning based FDS achieves an average of test AUC to 95.5%, which is about 10% higher than traditional FDS.

Keywords: Federated learning · Credit card fraud · Skewed dataset

1 Introduction

Credit card transactions take place frequently with the improvement of modern computing technology and global communication. At the same time, fraud is also increasing dramatically. According to the European Central Bank report [1], billions of Euros are lost in Europe because of credit card fraud every year.

K. Chen et al. (Eds.): BigData 2019, LNCS 11514, pp. 18–32, 2019.
https://doi.org/10.1007/978-3-030-23551-2_2

Credit card is considered as a nice target of fraud since a significant amount of money can be obtained in a short period with low risk [2]. Credit card frauds can be made in different forms, such as application fraud [3], counterfeit cards [4], offline fraud and online fraud [5]. Application fraud is a popular and dangerous fraud, it refers that fraudsters acquire a credit card by using false personal information or other person's information with the intention of never repaying the purchases [3]. Counterfeit fraud occurs when the credit card is used remotely; only the credit card details are needed [6]. Offline fraud happens when the plastic card was stolen by fraudsters, using it in stores as the actual owner while online fraud is committed via web, phone shopping or cardholder not-present [5].

There are two mechanisms that are widely used to combat fraud – fraud prevention and fraud detection. Fraud prevention, as the first line of defense, is to filter high risk transactions and stop them occurring at the first time. There are numerous authorization techniques for credit card fraud prevention, such as signatures [7], credit card number, identification number, cardholder's address and expiry data, etc. However, these methods are inconvenient for the customers and are not enough to curb incidents of credit card fraud. There is an urgent need to use fraud detection approaches which analyze data that can detect and eliminate credit card fraud [8].

However, there are many constraints and challenges that hinder the development of an ideal fraud detection system for banks. Existing FDS usually is prone to inefficient, with a low accuracy rate, or raises many false alarm, due to the reasons such as dataset insufficiency, skewed distribution and limitation of detection time.

- **Dataset Insufficiency**
  One of the main issues associated with the FDS is the lack of available public datasets [9]. The increasing concern over data privacy imposes barriers to data sharing for banks. At the same time, most fraud detection systems are produced in-house concealing the model details to protect data security. However, a reliable credit card FDS is impossible to be established in the absence of available dataset.
- **Skewed Distribution**
  Credit card transactions are highly unbalanced in every bank - where a few samples are fraud while a majority of them are legitimate transactions. In most circumstance, 99% of transactions are normal while fraudulent transactions are less than 1% [10]. In this case, it is very difficult for machine learning algorithms to discover the patterns in the minority class data. Furthermore, skewed class distribution has a serious impact on the performance of classifiers that are tend to be overwhelmed by the majority class and ignore the minority class [11].
- **Limitation of Detection Time**
  In some online credit card payment applications, the delay in time can lead to intolerable loss or potential exploitation by fraudsters. Therefore, an online FDS that has the ability to deal with limited time resource and qualifies enough to detect fraudulent activities rapidly is extremely important [12]. Building a good fraud detection framework which is fast enough to be utilized in a real-time environment should be considered.

In this paper, we aim to address these issues with a novel fraud detection system. First, we focus on a fraud detection system which can protect the data privacy, meanwhile, it can be shared with different banks. Then, we solve the problem of skewed distribution of datasets. A federated fraud detection framework with data balance approach is proposed to construct a fraud detection model, which is different from previous FDS. Federated fraud detection framework enables different banks to collaboratively learn a shared model while keeping all the training data which is skewed on their own private database. Furthermore, the accuracy, convergence rate, training time and communication cost of FDS are comprehensively taken into consideration.

The main contributions of this paper are summarized as follows:

(1) To deal with fraud detection problem and construct an effective FDS in data insufficient circumstance. A kind of decentralized data machine learning algorithm–federated fraud detection framework is proposed to train fraud detection model with the fraud and legitimate behavior features. Our work takes a step forward by developing ideas that solve the problem of dataset insufficiency for credit card FDS.
(2) Using the real-world dataset from the European cardholders, experiments are conducted to demonstrate our method is robust to the unbalanced credit card data distributions. Experimental results depicted that credit card FDS with federated learning improves traditional FDS[1] by 10% AUC and F1 score.
(3) From the results of experiments, conclusions that how to coordinate communication cost and accuracy of FDS are made, which would be helpful for making a trade off between computation resources and real-time FDS for future fraud detection work.

The rest of the paper is organized as follows. In Sect. 2, related work about credit card fraud is discussed. Section 3 gives the details of federated fraud detection framework. Section 4 provides an analysis of the dataset and experimental results. Conclusions and future work are presented in Sect. 5.

2 Related Work

Although fraud detection in the credit card industry is a much-discussed topic which receives a lot of attention, the number of public available works is rather limited [14]. One of the reasons is that credit card issuers protect the sharing of data source from revealing cardholder's privacy. In literature about credit card fraud detection, the data mining technologies used to create credit card FDS can categorized into two types: supervised method and unsupervised method.

Supervised learning techniques relies on the dataset that has been labeled as 'normal' and 'fraud'. This is the most prevalent approach for fraud detection. Recently, decision tree combined with contextual bandits are proposed to

[1] The traditional ensemble FDS [13] with SMOTE(borderline2) balancing techniques achieved AUC of 88% and F1 score of 82% on the same dataset.

construct a dynamic credit card fraud detection model [15]. Adaptive learning algorithms which can update fraud detection model for streaming evolving data over time [16] to adapt with and capture changes of patterns of fraud transactions. Data level balanced techniques such as under sampling approach [17], SMOTE and EasyEnsemble are conducted in [18] to find out the most efficient mechanism for credit card fraud detection. A supervised ensemble method [19] was developed by combining the bagging and boosting techniques together. Bagging technique used to reduce the variance for classification model through resampling the original data set, while boosting technique reduce the bias of the model for unbalanced data. A FDS constructed with a scalable algorithm BOAT (Boostrapped Optimistic Algorithm for Tree Construction) which supports several levels of the tree in one scan over the training database to reduce training time [8]. Other supervised learning methods in fraud are Bayes [20], artificial neural network(ANN) [21,22] and support vector machine [23,24].

In unsupervised learning, there is no class label for fraud detection model construction. As in [25], it proposed unsupervised methods that do not require the accurate label of fraudulent transactions but instead detect changes in behavior or unusual transactions. K-means clustering algorithm is an unsupervised learning algorithm for grouping a given set of data based on the similarity in their attribute used to detect credit card fraud [26].

The advantages of supervised FDS over semi-supervised and unsupervised FDS is that the outputs manipulated by supervised FDS are meaningful to humans, and it can be easily used for discriminative patterns. In this paper, a data level balance approach – SMOTE is used to handle the problem of skewed distribution by oversampling fruad transactions. Supervised method with a deep network (CNN) is applied by participated banks to detect fraud transactions. Federated fraud detection framework balances the FDS performance and training time by controlling deep network learning process. But one of the biggest differences is that fraud detection models described above are only trained by individual bank with whereas the model described in this paper is trained collaboratively by different banks.

3 Methodology

3.1 Preliminaries

This section formalizes the problem setting discussed in this paper, and the FFD framework.

Definition 1 (Transaction Dataset). 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.

Definition 2 (Loss Function). 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 $\ell(x_i, y_i; w)$.

Definition 3 (Learning Rate). The learning rate controls the speed that model converges to the best accuracy. We define the learning rate as η.

Machine learning algorithm always centralizes the training data on a data center. Credit card transaction information is sensitive in both its relationship to customer privacy and its importance as a source of proprietary information for banks. Traditional machine learning models for credit card fraud detection are typically trained by individual banks with their own private dataset. Due to these datasets are privacy sensitive and large in quantity, federated learning was presented by Google [27] in 2017. Different from the traditional machine learning, federated learning enables to collaboratively learn a shared model. This shared model is trained under the coordination of a central server by using dataset distributed on the participating devices and default with privacy [28]. A typical federated algorithm – FederateAveraging (FedAvg) algorithm based on deep learning was introduced. FedAvg algorithm combines local stochastic gradient descent (SGD) on each client with a central server that performs model averaging [29]. Each client is used as nodes performing computation in order to update global shared model maintained by the central server. Every client trained their own model by using local training dataset which is never uploaded to the central server, but the update of model will be communicated. Federated learning can be concluded to five steps [27]: (1) Participating device downloads the common model from the central server. (2) Improving the model by learning data on local device. (3) Summarizes the changes of the model as a small focused update and send it using encrypted communication to the central server. (4) The server immediately aggregates with

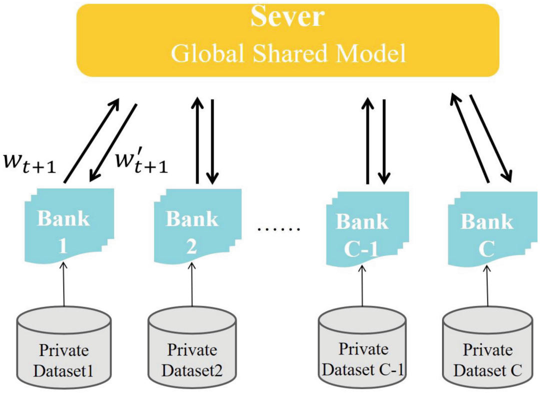


Fig. 1. Diagram of the Federated learning Framework. w_{t+1} represents the banks parameter that upload to server, w'_{t+1} represents the parameter that averaging by server.

other device updates to improve the shared model. 5) The process repeats until convergence. The structure of federated learning is illustrated in Fig. 1.

3.2 Federated Fraud Detection Framework

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,..,C). x_i^c is the feature vector, y_i^c is the corresponding label and n_c is the size of dataset associated with participant bank c. Credit card transaction data is skewed, fraudulent transactions have a very small percentage of total number of dataset, which might cause obstructions to the performance of credit card FDS. A data level method–SMOTE [30] is selected for data rebalancing at D_i. SMOTE oversamples the minority class by generating synthetic minority examples in the neighborhood of observed ones. It is easier to implement and does not lead to increase training time or resources compared to algorithm level approach [18].

In our fraud detection system with federated learning, the goal is to allow 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). For a non-convex neural network model objective is:

$$\min_{w \in \mathbb{R}^d} \quad \ell(x,y;w) \qquad where \qquad \ell(x,y;w) \overset{def}{=} \frac{1}{n}\sum_{i=1}^{n} \ell(x_i,y_i;w). \tag{1}$$

In federated fraud detection model, There are C banks as participant with a fixed dataset $|D_i| = n_c$, We use n to represent all the data samples involved in the whole FDS. Thus n= $\sum_{i=1}^{C} |D_i|$ =$\sum_{c=1}^{C} n_c$. We can re-write the objective (1) as

$$\ell(x,y;w) = \qquad where \qquad L_c(x_c,y_c;w) = \frac{1}{n_c}\sum_{i \in D_i} \ell(x_i^c, y_i^c; w) \tag{2}$$

The server will initialize the fraud detection model parameters. At each communication round t=1,2,...., a random fraction F 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 computing the average gradient of the loss f_c on their own private dataset at current fraud detection model parameters w_t with a fixed learning rate η, $f_c = \nabla L_c(x_c, y_c; w)$. These banks update their fraud detection model synchronously and send the update of fraud detection model to server.

The server aggregates these updates and improves the shared model

$$w_{t+1} \leftarrow w_t - \eta \nabla \ell(x,y;w) \tag{3}$$

$$w_{t+1} \leftarrow w_t - \eta \sum_{c=1}^{C} \frac{n_c}{n} \nabla L_c(x_c, y_c; w) \tag{4}$$

$$w_{t+1} \leftarrow w_t - \eta \frac{n_c}{n} f_c \tag{5}$$

For every bank c, $w_{t+1}^c \leftarrow w_t - \eta f_c$, since (5), then

$$w_{t+1} \leftarrow w_t - \sum_{c=1}^{C} \frac{n_c}{n} w_{t+1}^c \tag{6}$$

Considering the impact of skewed data on model performance, we use the combination of data size and detection model performance α_{t+1}^c on each bank as the weight of parameter vector. it can be written as

$$w_{t+1} \leftarrow w_t - \sum_{c=1}^{C} \frac{n_c}{n} \alpha_{t+1}^c w_{t+1}^c \tag{7}$$

Increasing the weight of strong classifiers and make it plays a more important role to form a better global shared model. Each bank takes a step of gradient descent and evaluates on fraud detection model using its own credit card transactions. Then, the server applies them by taking a weighted average and makes them available to all participated banks. The whole process will go on for T iterations.

Algorithm 1. FFD framework. The C banks are index by n; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

```
Input: The private dataset of banks and financial institutions
Output: A credit card fraud detection model with federated learning
  ServerUpdate :
    initialize the detection classifier and its parameters $w_0$
    for each round t=1,2,...T do:
      Random choose max(F*C, 1) banks as $N_t$
      for each banks c $\in N_t$ in parallel do
        $w_{t+1}^c$, $\alpha_{t+1}^c \leftarrow \boldsymbol{BankUpdate(n, w_t)}$
      $w_{t+1} \leftarrow \sum_{t=1}^{T} \frac{n_c}{n} \alpha_{t+1}^c w_{t+1}^n$

  BankUpdate(n, w) :
    Data processing: rebalance raw dataset with SMOTE and split them into two
    part: 80% training data and 20% testing data
    Training:
      $B \leftarrow$ split $D_n$ into baches of size B
      for each local epoch i from l to E do
        for batch b $\in B$ do
          w$\leftarrow$ w - $\eta \nabla \ell$(x,y;w)
    Testing
    return w and validation accuracy $\alpha$ to server
```

The increasing concern over data privacy imposes restrictions and barriers to data sharing and make it difficult to coordinate large-scale collaborative constructing a reliable FDS. Credit card FDS based on federated learning is proposed, it enables each bank to train a fraud detection model from data distributed across multiple banks. It not only helps credit card FDS learn better patterns of fraud and legitimate transactions but also protect the datasets' privacy and security. For federated optimization, communication cost is a major challenge. On the one hand, banks should fetching initial fraud detection model parameters from server. At the same time, banks should upload the update of model to server. So communication cost in FDS is symmetric. It is influenced by upload bandwidth, but in our FDS, the communication cost is related three key parameters: F, the fraction of banks that be selected to perform computation on each round; B, the minibatch size used for banks update. E, the number of local epochs. Controlling communication cost by tuning these three parameters which means we can add computation by using more banks to participate to increase parallelism or performing more computation on each bank between every communication round. The details of our fraud detection model training process are described in Algorithm 1.

4 Experimental Results

This section is organized as three parts. Firstly, we introduce the dataset that used in our FDS. Secondly, we show the performance measurement of our fraud detection model. Finally, we demonstrate the results of our experiments.

4.1 Dataset Description

We conducted a series of comprehensive experiments to manifest the superiority of our method. The experiment dataset from the European Credit Card (ECC) transactions made in September 2013 by European cardholders and it provided by the ULB ML Group [31]. This dataset contains anonymized 284,807 total transactions spanning over a period of two days, but there are only 492 fraudulent transactions in this dataset with a ratio of 1:578. The dataset is highly imbalanced as it has been observed only 0.172% of the transactions are fraudulent. Due to confidentiality issues, the original features, and some background information about the dataset cannot be provided. So this dataset contains only 30 numerical input variables which are a result of the Principal Component Analysis(PCA) transformation. It is described in Table 1. This is a classic example of an unbalanced dataset of credit card fraud(Fig. 2), it is very necessary to rebalance the raw data to prevent the classifiers from over-fitting the legitimate class and ignore the patterns of frauds.

Table 1. Credit card dataset

Normal	Fraud	Features	Instance
284315	492	30	284807

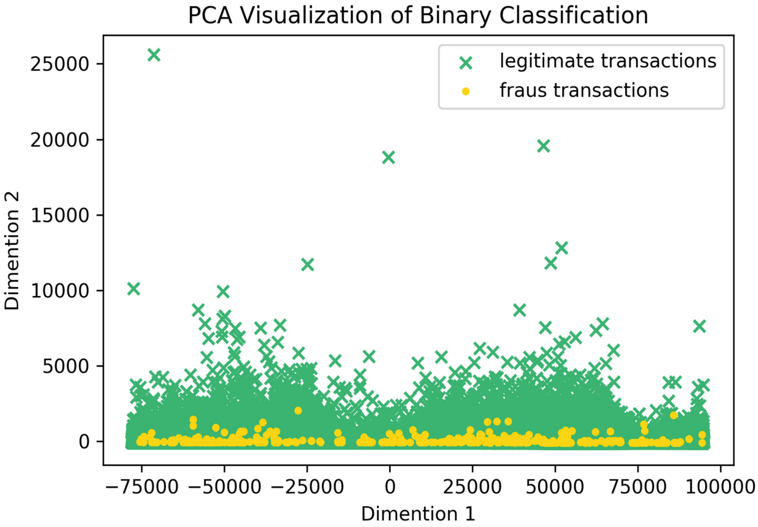


Fig. 2. Dataset visualization via PCA.

4.2 Performance Measures

Measuring the success of machine learning algorithm is a crucial task so that the best parameters suitable for credit card fraud detection system can be selected [32]. When the dataset is significantly imbalanced, accuracy is not enough to measure the performance of FDS. Accuracy will have a high value even if the FDS mispredict all instances to legitimate transactions. Therefore, we take other measures into consideration namely precision, recall, F1 and AUC which are calculated based on Table 2 where Positive correspond to fraud samples and Negative correspond to legitimate samples. Accuracy indicates the total experimental records have been classified correctly by FDS. Precision rate is a measurement of reliability of the FDS while recall rate measures the efficiency of FDS in detecting all fraudulent transactions. F1 is the harmonic mean of recall and precision. Additionally, Area Under Curve(AUC) refers to the area under the Receiver Operating Characteristic(ROC) curve, which can better describe the performance of classifiers trained with unbalanced samples.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

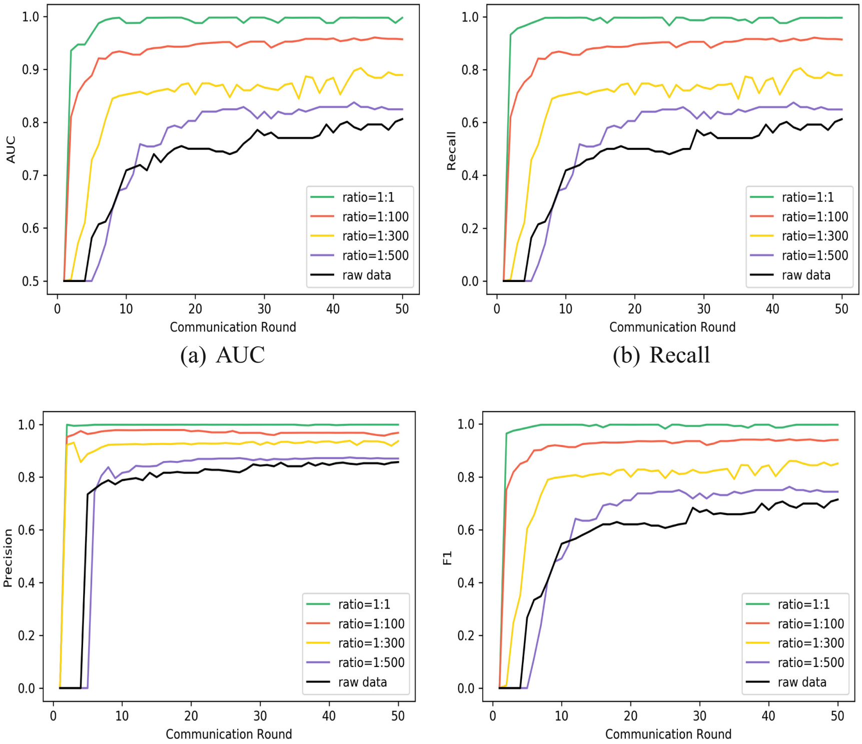
$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

Table 2. Performance matrix

Predit	Real	
	Positive(Fraud)	Negative(Normal)
Positive(Fraud)	True Positive (TP)	False Positive (FP)
Negative(Normal)	False Negative (FN)	True Negative (TN)

4.3 Results and Discussions

In this section, a series of experiments are conducted to show the advancement of our fraud detection system. All the experiments are running on a standard server on Intel E5 with 28 CPU cores, 2.00 GHz, and 128 GB RAM. The shared global model is a CNN [33] with two convolution layers, the first with 32 channels and the second with 64 channels, each layer followed with a max pooling, a fully connected layer with 512 units and RELU activation, and a final softmax output layer.



(a) AUC

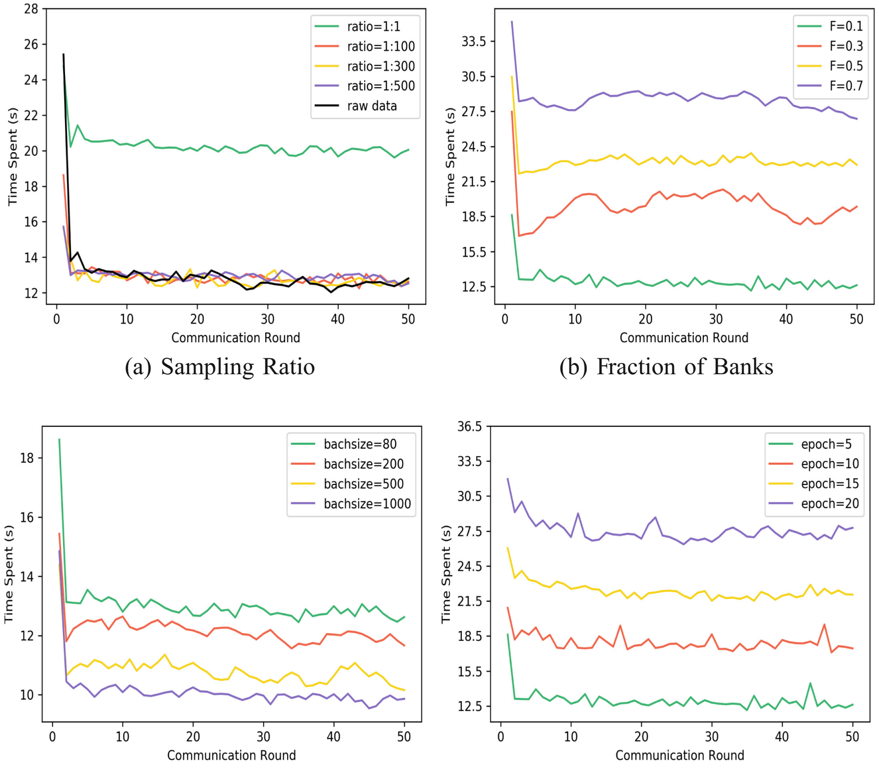
(b) Recall

(c) Precision

(d) F1

Fig. 3. Sensitive test of sampling ratio of fraud and legitimate transactions.

To minimize the impact of over-fitting, we split the dataset into 80% training data and 20% testing data. Data level approach–SMOTE is selected to rebalance raw dataset. We conduct a series of experiments on different sampling ratio with a default E = 5, B = 80 and $\eta = 0.01$. Figure 3 shows that the federated FDS with data balance mechanism outperforms FDS that trained with raw data. The better fraud detection system performed with a higher proportion of fraud transactions. Due to FDS can learn better patterns of fraud and legitimate transaction when the data is more balance. Figure 4(b) depict that when the sampling ratio is 1:1 which refers to the ratio of fraud and legitimate transactions over 1:1, the training time increased sharply but there is only a small advantage to FDS performance. Taking the training time and realistic application into consideration, we choose the sampling ratio of 1:100 to achieve an efficient FDS. From real business perspective, the average cost of misjudging 100 normal transactions is approximately the same as the mean cost of missing a fraudulent transaction.



(a) Sampling Ratio

(b) Fraction of Banks

(c) Batchsize

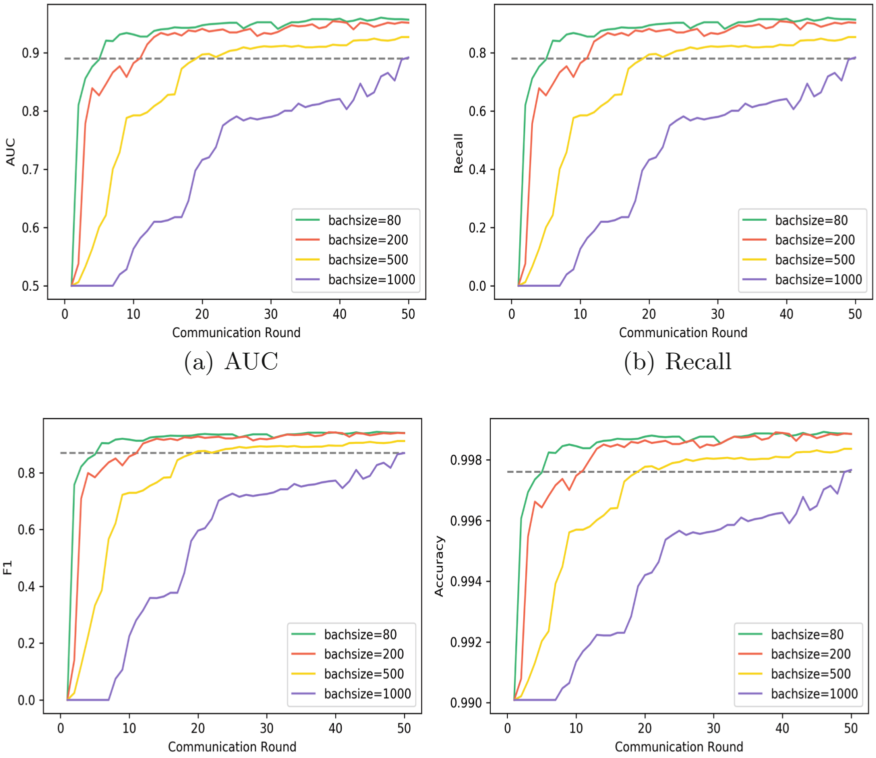
(d) Epochs

Fig. 4. Efficiency of federated FDS.

After rebalancing the dataset, we also should specify the data distribution on each bank. The dataset is shuffled, and then partitioned into C = 100 banks randomly. Because the amount of transactions owned by each bank is different in reality, each bank receives a different amount of transactions. Then, the experiments with the fraction of banks F which controls the amount of banks parallelism are implemented. Table 3 demonstrates the impact of varying F for credit card fraud detection system. We calculate the number of communication round to reach a target AUC of 95.9%. The first line of Table 3 demonstrates that with the increasing Banks involved in parallel computing, the number of communication round required to reach the target AUC decreased, but the performance of FDS has become better. Time efficiency is also essential to an effective FDS which should be able to deal with limited time resource. In our FDS, the training time of every communication round (Fig. 4(a)) shows an improvement in increasing fraction of banks, but there is small advantages in performance. In order to keep a good balance between the performance of FDS and computational efficiency, in remainder experiments, we fixed $F = 0.1$.

Table 3. Sensitive test of fraction of banks

	$F = 0.1$	$F = 0.3$	$F = 0.5$	$F = 0.7$
Communication rounds	35	30	28	18
Best AUC	0.9555	0.9603	0.9638	0.9690
Best F1	0.9393	0.9441	0.9448	0.9534
Time/round(s)	12.94	19.45	23.27	28.53



(a) AUC

(b) Recall

(c) F1

(d) Accuracy

Fig. 5. Sensitive test of local batch size.

With $F = 0.1$, adding more computation per bank on each round by decreasing batch size or increasing epochs. For batch size –B, we calculate the number of communication rounds necessary to achieve a target recall of 78%, F1 of 87%, AUC of 89% and validation accuracy of 99%. The results are depicted in Fig. 5, where the grey lines stand for the targets. In Fig. 4(c), smaller batch size lead longer training time on average. But the number of communication rounds to reach the targets is decreased with the increasing computation per bank by decreasing the local batch size of banks. So the total time cost is still decreased. Smaller batch size speeds the convergence and improves the performance of FDS. For local epochs, Fig. 4(d) shows that larger epoch leads the increment of training time to per communication round. But Table 4 depicts that the number to reach the target AUC of 96% is decreased. Figure 5 and Table 4 reveal that add more local SGD updates by decreasing batch size or increasing epochs per round to each bank result in a speed up to convergence rate and less computation cost.

Table 4. Sensitive test of number of local epochs

	$E = 5$	$E = 10$	$E = 15$	$E = 20$
Communication rounds	46	23 (0.5×)	15 (0.33×)	8 (0.17×)
Time/round(s)	12.98	17.96	22.38	27.56
Total Time(s)	596.08	413.08	335.7	220.48

5 Conclusion

This paper constructed a credit card FDS with federated detection. The results of our experiments show that federated learning for credit card detection system has a significant improvement. Federated fraud detection framework enables banks without sending their private data to data center to train a fraud detection system. This decentralized data method can protect the dataset sensitivity and security, alleviate the influence of unavailable dataset to some degree. There are still privacy problems in federated fraud detection system. First, we should consider what information can be learned by inspecting the global shared model parameters. Second, we should think about what privacy-sensitive data can be learned by gaining access to the updates of an individual bank. In future works, we will take more reliable measurements into account to protect the privacy of data. And the Non-IID dataset can be evaluated in this credit card fraud detection system and ensure the credit card FDS to communicate and aggregate the model updates in a secure, efficient and scalable way.

Acknowledgment. This work is supported by China National Basic Research Program (973 Program, No. 2015CB352400), National Natural Science Foundation of China (No. 61572488,61572487), Shenzhen Basic Research Program (No. JCYJ201803021457 31531, JCYJ20170818163026031), and Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence.

References

1. Bahnsen, A.C., Aouada, D., Stojanovic, A., Ottersten, B.: Feature engineering strategies for credit card fraud detection. Expert Syst. Appl. **51**, 134–142 (2016)
2. Zareapoor, M., Shamsolmoali, P., et al.: Application of credit card fraud detection: based on bagging ensemble classifier. Procedia Comput. Sci. **48**(2015), 679–685 (2015)
3. Bolton, R.J., Hand, D.J.: Statistical fraud detection: a review. Stat. Sci., 235–249 (2002)
4. Sahin, Y., Bulkan, S., Duman, E.: A cost-sensitive decision tree approach for fraud detection. Expert Syst. Appl. **40**(15), 5916–5923 (2013)
5. Laleh, N., Abdollahi Azgomi, M.: A taxonomy of frauds and fraud detection techniques. In: Prasad, S.K., Routray, S., Khurana, R., Sahni, S. (eds.) ICISTM 2009. CCIS, vol. 31, pp. 256–267. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00405-6_28
6. Delamaire, L., Abdou, H., Pointon, J., et al.: Credit card fraud and detection techniques: a review. Banks Bank Syst. **4**(2), 57–68 (2009)
7. Abdallah, A., Maarof, M.A., Zainal, A.: Fraud detection system: a survey. J. Netw. Comput. Appl. **68**, 90–113 (2016)
8. Sherly, K., Nedunchezhian, R.: Boat adaptive credit card fraud detection system. In: 2010 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–7. IEEE (2010)
9. Jha, S., Guillen, M., Westland, J.C.: Employing transaction aggregation strategy to detect credit card fraud. Expert systems with applications, 39(16), 12650–12657 (2012)
10. Bahnsen, A.C., Stojanovic, A., Aouada, D., Ottersten, B.: Improving credit card fraud detection with calibrated probabilities. In: Proceedings of the 2014 SIAM International Conference on Data Mining, pp. 677–685. SIAM (2014)
11. Liu, X.-Y., Wu, J., Zhou, Z.-H.: Exploratory undersampling for class-imbalance learning. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **39**(2), 539–550 (2009)
12. Minegishi, T., Niimi, A.: Proposal of credit card fraudulent use detection by online-type decision tree construction and verification of generality. Int. J. Inf. Secur. Res. (IJISR) **1**(4), 229–235 (2011)
13. Mohammed, R.A., Wong, K.-W., Shiratuddin, M.F., Wang, X.: Scalable machine learning techniques for highly imbalanced credit card fraud detection: a comparative study. In: Geng, X., Kang, B.-H. (eds.) PRICAI 2018. LNCS (LNAI), vol. 11013, pp. 237–246. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97310-4_27
14. Van Vlasselaer, V., et al.: APATE: a novel approach for automated credit card transaction fraud detection using network-based extensions. Decis. Support Syst. **75**, 38–48 (2015)
15. Soemers, D.J., Brys, T., Driessens, K., Winands, M.H., Nowé, A.: Adapting to concept drift in credit card transaction data streams using contextual bandits and decision trees. In: AAAI (2018)
16. Žliobaitė, I.: Learning under concept drift: an overview. arXiv preprint arXiv:1010.4784 (2010)
17. Chen, R.-C., Chen, T.-S., Lin, C.-C.: A new binary support vector system for increasing detection rate of credit card fraud. Int. J. Pattern Recogn. Artif. Intell. **20**(02), 227–239 (2006)

18. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., Bontempi, G.: Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst. Appl. **41**(10), 4915–4928 (2014)
19. Bian, Y., et al.: Financial fraud detection: a new ensemble learning approach for imbalanced data. In: PACIS, p. 315 (2016)
20. Bahnsen, A.C., Stojanovic, A., Aouada, D., Ottersten, B.: Cost sensitive credit card fraud detection using bayes minimum risk. In: Proceedings-2013 12th International Conference on Machine Learning and Applications, ICMLA 2013, vol. 1, pp. 333–338. IEEE Computer Society (2013)
21. Patidar, R., Sharma, L., et al.: Credit card fraud detection using neural network. Int. J. Soft Comput. Eng. (IJSCE) **1**, 32–38 (2011)
22. Syeda, M., Zhang, Y.-Q., Pan, Y.: Parallel granular neural networks for fast credit card fraud detection. In: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2002, vol. 1, pp. 572–577. IEEE (2002)
23. Lu, Q., Ju, C.: Research on credit card fraud detection model based on class weighted support vector machine. J. Convergence Inf. Technol. **6**(1) (2011)
24. Wu, C.-H., Tzeng, G.-H., Goo, Y.-J., Fang, W.-C.: A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. Expert Syst. Appl. **32**(2), 397–408 (2007)
25. Bolton, R.J., Hand, D.J., et al.: Unsupervised profiling methods for fraud detection. Credit Scoring and Credit Control VII, pp. 235–255 (2001)
26. Srivastava, A., Kundu, A., Sural, S., Majumdar, A.: Credit card fraud detection using Hidden Markov Model. IEEE Trans. Dependable Secure Comput. **5**(1), 37–48 (2008)
27. McMahan, B., Ramage, D.: Federated learning: Collaborative machine learning without centralized training data. Google Research Blog (2017)
28. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency, arXiv preprint arXiv:1610.05492 (2016)
29. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., et al.: Communication-efficient learning of deep networks from decentralized data, arXiv preprint arXiv:1602.05629 (2016)
30. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
31. ccfraud dataset. https://www.kaggle.com/mlg-ulb/creditcardfraud
32. West, J., Bhattacharya, M.: Some experimental issues in financial fraud mining. In: ICCS 2016, pp. 1734–1744 (2016)
33. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)