SN Computer Science (2024) 5:578
https://doi.org/10.1007/s42979-024-02898-y

ORIGINAL RESEARCH

A Comprehensive Fraud Detection for Credit Card Transactions in Federated Averaging

Tran Khanh Dang[1] · Trung Ha[2]

Received: 13 November 2023 / Accepted: 10 April 2024

Abstract

Credit card fraud costs card issuers billions of dollars each year. Therefore, an effective Fraud Detection System (FDS) is crucial to minimize losses for banks and cardholders. Based on real data collected by the European Credit Card, the dataset is highly skewed, with the number of fraudulent samples significantly lower than legitimate transactions. Due to data security and privacy concerns, different banks are generally not allowed to share their transaction datasets. These challenges not only make it difficult to build traditional credit card fraud detection models but also to detect fraud effectively. In this paper, we evaluate federated learning for fraud detection in identifying illegitimate transactions. Unlike traditional models trained on centralized cloud data, federated learning models allow banks to train fraud detection models using their own local databases. Subsequently, a shared global model is constructed by aggregating locally computed updates of the fraud detection model. This approach enables banks to collectively benefit from a collaborative model without sharing datasets and safeguards cardholders' sensitive information. Experimental results demonstrate that FDS, based on federated learning, achieved an accuracy rate of up to 97% on the European Credit Card (ECC) transactions dataset after over 60 training rounds.

Keywords Federated learning · Imbalance data · Credit card fraud detection · Horizontal federated learning

Introduction

In recent years, with the popularity of credit cards and the rapid development of electronic services such as e-commerce, e-finance, mobile payments, the volume of credit card transactions has been increasing sharply. The broad use of credit cards and diverse transaction scenarios without strict verification and monitoring will necessarily lead to billions of dollars in losses due to credit card fraud. It is difficult to get an accurate estimate of the loss because cardholders are often reluctant to release statistics. According to a report by the European Central Bank [1], billions of Euros were lost in Europe as a result of credit card fraud every year. Credit cards were considered a good target of fraud because an important amount of money can be secured in a short time period with low risk [2]. Credit card fraud can take many different forms such as application fraud [3], card forgery [4], offline fraud and online fraud [5]. Application fraud is a common and dangerous form of fraud, it refers to fraudsters obtaining a credit card by using false personal information with the intention of never refunding the purchases [3]. Counterfeit fraud occurs when the credit cards are used remotely; only the credit card details are needed [6]. Offline fraud takes place when fraudsters steal a plastic card and use it in stores as the real owner while online fraud is accomplished via the web, shopping by phone or cardholder is not present [5].

There are two widely used mechanisms to combat fraud such as fraud prevention and fraud detection. The first mechanism is fraud prevention, which involves filtering high-risk transactions and blocking them from occurring initially [7]. There are many authorization techniques to prevent credit card fraud, such as signatures [8], credit card numbers,

This article is part of the topical collection "Future Data and Security Engineering 2022" guest edited by Tran Khanh Dang.

✉ Trung Ha
trunghlh@uit.edu.vn

Tran Khanh Dang
khanh.dt@vlu.edu.vn

[1] Faculty of Information Technology, Van Lang University, 69/68 Dang Thuy Tram, Ward 13, Binh Thanh District, Ho Chi Minh City, Vietnam

[2] University of Information Technology, VNU-HCM, Thu Duc City, Vietnam

Published online: 23 May 2024

identification number, cardholder's addresses, and expiration date, etc. [9]. However, these methods are complicated for the customer and are insufficient to limit incidents of credit card fraud. There is an urgent need to employ fraud detection approaches to analyze data that can detect and eliminate credit card fraud [10]. Secondly, credit card fraud detection is also an important way to prohibit fraudulent events, which is generally classified into two approaches such as anomaly detection [11] and classifier-based detection [12] as Fig. 1 shows.

To combat credit card fraud, financial services companies use a variety of techniques, including machine learning algorithms, to detect fraudulent transactions. Machine learning algorithms can analyze large amounts of data to identify patterns that may indicate fraud [13]. However, the use of machine learning algorithms for credit card fraud detection presents its own set of challenges, such as the need to balance accuracy and speed with the need to protect customer privacy. Federated learning, which enables collaborative training while preserving privacy, can help address these challenges.

The main contributions of this study are as follows: first, to deal with fraud detection problems and construct an effective FDS in data insufficient situations. A kind of decentralized data machine learning algorithm–federated fraud detection framework is proposed to train a fraud detection model with the fraud and legitimate behavior features. The second contribution is to design a specific federated learning scheme, which has achieved a similar effect of centralizing bank data for modeling. In practice, it has been proved that federated learning can not only protect privacy data but also achieve a better risk identification effect, which strongly supports such typical applications. The third contribution is the real-world dataset from the European cardholders' experiments conducted to demonstrate our method. The experimental results depicted that credit card FDS with federated learning improves traditional FDS by 10% AUC and F1 score. In this study, it is assumed that the federated learning model used in the research is client–server, with banks as clients and government organization, or trusted third party as the central server.

The remainder of this paper is organized as follows. "Related work" section discusses the related work to credit card fraud detection, then "Background" section describes theory and methods. "The experimentation results" section details the evaluation setup and the conclusions and future work are provided in "Conclusion and future work" section.

Related Work

Although fraud detection in the credit card industry is a topic of much discussion and interest, the number of published research papers is rather limited [14]. One of the reasons is that credit card issuers protect shared data from disclosing cardholder's privacy. In the credit card fraud detection system, the data mining technologies used to implement credit card FDS are mainly towards centralized machine learning architecture which can be classified into two categories: supervised methods and unsupervised methods.

Supervised learning techniques are based on datasets that have been labeled as 'normal' and 'fraud'. This is the most common approach to fraud detection as shown in Table 1. Recently, decision trees combined with contextual data point anomaly detection have been proposed to build a dynamic credit card fraud detection and credit risk scoring model [15]. Adaptive machine learning algorithms can update the fraud detection model from receiving time-evolving data to adapt and capture changes in fraudulent transaction patterns. Data-level balancing techniques, such as sampling [16] and resampling [17] approach, have been implemented by Dal Pozzolo et al. They utilized SMOTE (synthetic minority over-sampling technique) and EasyEnsemble [18] to find the most effective credit card fraud detection mechanism. In [19], the data imbalance problem was solved and compared four machine learning algorithms (decision tree, random forest,

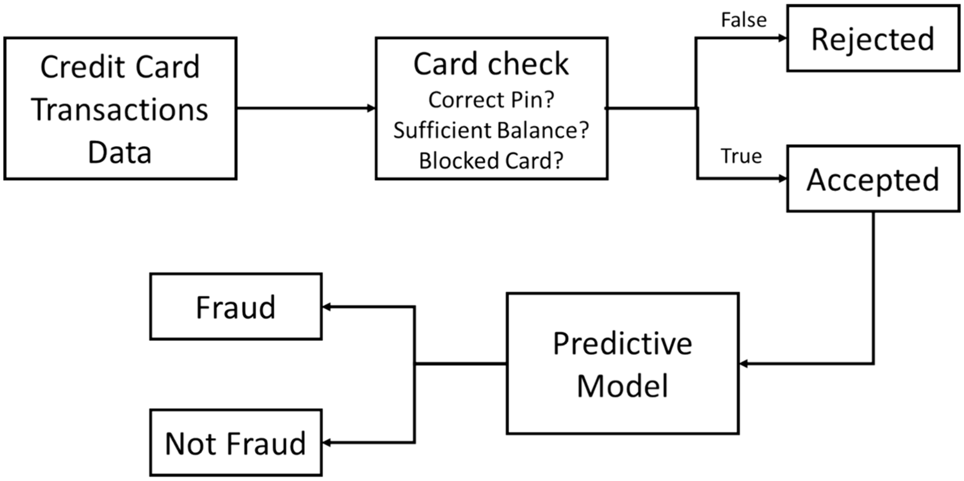


Fig. 1 Credit card fraud detection process

Table 1 Credit card fraud detection methods and types

Method	Type	Description	Advantage
Anomaly detection [11]	Statistical analysis	Identify transactions that deviate from typical patterns using statistical methods	Can detect unknown fraud patterns
Decision trees [15]	Supervised learning	Build decision trees to classify transactions as fraudulent or legitimate based on various features	Be easy to interpret and explain
Random forest [21]	Ensemble learning	Build multiple decision trees and aggregates their predictions to improve accuracy	Can handle complex fraud patterns
Clustering [23]	Unsupervised learning	Group similar transactions together and flags outliers	Can identify unknown fraud patterns
Neural networks [24]	Deep learning algorithms	Train deep learning models to detect fraudulent patterns in transaction data	Can detect complex fraud patterns and reduce false positives
Autoencoder [25]	Deep learning algorithms	Train a neural network to reconstruct input transactions and flag those with high reconstruction errors	Can detect complex fraud patterns and reduce false positives
Gradient boosting [20]	Ensemble learning	Build multiple decision trees sequentially, with each tree improving on the previous one's errors	Can handle complex fraud patterns and reduce false positives
Logistic regression [26]	Supervised learning	Fit a logistic regression model to predict the probability of a transaction being fraudulent	Be simple and interpretable

logistic regression, and k nearest neighbors) to evaluate the performance of the obtained balanced dataset. The results showed better performance with a balanced dataset than with a skewed imbalance dataset. A supervised aggregation method [20] has been developed by combining the advantages of both bagging and boosting techniques. The bagging technique is used to reduce the variance for the classification model through resampling the original data set, while the boosting technique is reducing the bias of the model for unbalanced data [21]. An FDS built using the BOAT algorithm (Boostrapped Optimistic Algorithm for Tree Construction) which supports several levels of the tree in one scan through the training database to reduce training time [10].

In unsupervised learning, there are no class labels for building a fraud detection model. As in [22], it proposes unsupervised methods that do not require exact labels of fraudulent transactions but instead detect changes in behavior or unusual transactions. Clustering algorithm is an unsupervised learning algorithm to group a given set data based on the similarity in their attributes used for credit card fraud detection [23]. The output of an anomaly detection method can be either directly a label (anomaly or normal data), or a score that measures the degree of anomaly, which is then normally compared to a threshold to arrive at a decision [11].

The advantage of supervised FDS over semi-supervised and unsupervised FDS is that the outputs controlled by the supervised FDS are significant to humans, and it can easily be used for discriminant patterns. Besides traditional machine learning methods, deep learning methods are also used to detect credit card fraud such as a deep network applied by participating banks to detect fraud transactions [24]. In addition, deep learning algorithms (Autoencoders) [25] are used to detect fraud and it has been found that the produced results are better than gradient boosted trees and logistic regression [26]. The federated fraud detection framework balances FDS performance and training time by controlling the deep network learning process. But one of the biggest differences is that the fraud detection models described previously are only trained by individual banks whereas the model described in this paper is trained by different banks, then performed the model parameter exchanges to increase accuracy.

In recent years, federated Learning has become a powerful tool for training data models from a large number of devices, making it ideal for applications in a variety of fields. These include sectors such as healthcare, finance, and connected vehicles. In these systems, federated learning allows training a central model while keeping user data private on each device. However, ensuring high model accuracy requires participation of a significant number of reliable users. Designing a fair incentive system is crucial for attracting these users. Challenges arise due to varying resources and user contributions. A poorly designed system can lead to “free riding” where some users benefit without contributing, ultimately harming the accuracy of the model. Dang et al. [27] combined contract theory and shapley value to create fair incentives based on each user's actual contribution to the model, promoting high-quality participation. Additionally, Pham et al. [28] proposed collaborative filtering suitable for the task of ranking, protecting user interaction information through neural collaborative filtering and Bayesian personalized ranking. Notably, the authors' approach minimizes communication overhead, ensuring efficient use of user data while preserving privacy.

Background

Federated Learning

FL is a robust developing machine learning, to solve the problem of lacking data to increase the accuracy of machine learning models while preserving data privacy. It refers to multiple clients (such as mobile devices, organizations, etc.) coordinated with one or more central servers for the decentralized machine learning settings. This approach was first introduced by Google in 2016 to predict a user's text input across hundreds of thousands Android devices while ensuring that data remains on the user's devices [29]. The process of FL is generally depicted as shown in Fig. 2. This type of federated learning is known as federated average (Feeding), which is the baseline of FL in different researchers. Firstly, each device downloads a global model for the subsequent local training. Secondly, the downloaded global model is refined through multiple local updates with individual device data, each belonging to different mobile devices. These devices then upload the corresponding gradient information to the central server. Thirdly, the averaged updates of local models implemented on the central server are distributed to the devices as an updated global model. Finally, the above functions repeat until the model achieves a certain desired performance or the final round arrives. The appearance of this technology will solve the conflict between data privacy and data sharing for various devices. Due to the property that data is not revealed to the third central server, FL is appropriate for application when data is a type of sensitive personal data.

More formally, consider N data owners $\{P_i\}_{i=1}^{N}$ who participate to train a ML model by using their respective datasets $\{D_i\}_{i=1}^{N}$. A conventional approach is to collect all data $\{D_i\}_{i=1}^{N}$ together at one data server and train a ML model M_{SUM} on the server using the centralized dataset. In the conventional approach, any data owner $\{P_i\}$ will reveal its data $\{D_i\}$ to the server and even other data owners.

Federated learning is a ML process in which the data owners collaboratively train a model M_{FED} without collecting all data $\{D_i\}_{i=1}^{N}$. Denote V_{SUM} and V_{FED} as the performance measure of the centralized model M_{SUM} and the federated model M_{FED}, respectively. Let δ be a non-negative real number. The federated learning model M_{FED} has δ-performance loss if

$$\left|V_{SUM} - V_{FED}\right| < \boldsymbol{\delta} \tag{1}$$

Equation (1) expresses the following intuition: if federated learning is used to build an ML model on distributed data sources, the performance of this model on future data is approximately the same as the model built by combining all data sources.

Categories of Federated Learning

Let matrix D_i represent the data held by the Ith data owner. Suppose that each row of the matrix D_i is a data sample, and each column is a specific feature. At the same time, some datasets may also contain label data. The data is denoted as follows: the feature space as X, the label space as Y, and the sample ID space as I. For example, in the financial field, labels may be users' credit. In the ecommerce field, labels may be the user's purchasing item. The feature X, label Y, and sample IDs I constitute the complete training

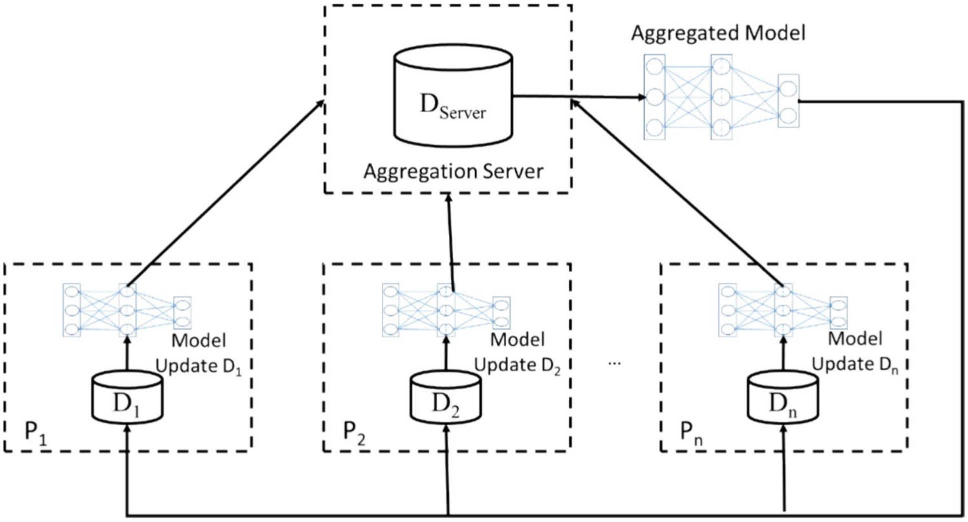


Fig. 2 A federated learning architecture: the client–server model

dataset (I;X;Y). The dataset's feature and sample spaces of the participants may not be identical. Based on how data is partitioned among various parties in the feature and sample ID space [30], FL largely falls into three groups, respectively, horizontal federated learning (HFL), vertical federated learning (VFL), and federated transfer learning (FTL).

Horizontal Federated Learning

In the case of horizontal FL, there is a certain amount of overlap between the features of data spread across various participants, while the data are quite different in sample space. At present, the existing FL algorithms primarily aim at application in smart devices or devices on the Internet of Things (IoT). FL in these scenarios usually could classify into horizontal FL. Because data may significantly differ in sample space but have similar feature space simultaneously. As mentioned above, the federated model solution for Android mobile phone update raised by Google [29] is typically a kind of horizontal FL since the data has the same feature dimension. In real applications such as finance, a large amount of work is inseparable from data collection. When it comes to cross-regional cooperation, it is almost impossible for each bank to build a data pool for sharing. Thus, FL could construct a federal network for cross-regional banks with similar loan and credit information to improve joint models as shown in Table 2.

Vertical Federated Learning

Vertical FL is suitable for cases in which data partitioned into the vertical direction according to feature dimension. All the parties hold homogeneous data which has partial overlap on sample ID whereas differ in feature space. For example, there is a financial institution, and they intend to identify fraud transactions, such as credits and loans in a predictive way. According to research, transactions which suffer from the abnormal amount and type of transaction may be identified as high risk [31]. Therefore, it can be analyzed in view of some rough dimensions, such as customers' age and loan as well as financial history. If a transaction occurs with a normal amount and type but is the first receipt at a supermarket, it may still be prone to fraud. However, predicting and personalizing such transactions is challenging due to the lack of information. With the development of FL, collaboration with retail companies holding purchase information becomes feasible. Furthermore, they can cooperate with each other without the need for raw data transmission, as shown in Table 3. Generally, scholars deal with this problem through taking out the same entities with various characteristics to get joint training. Compared to horizontal FL, vertical FL presents greater challenges due to entity resolution [32]. Unlike horizontal FL, aggregating all datasets onto a common server to learn from a global model does not work for vertical FL. This is because addressing the correspondence between different owners remains an urgent need [33].

Transfer Federated Learning

Unlike horizontal FL and vertical FL scenarios, in most cases, data neither shares sample space nor feature space. The main challenge in this setting is the lack of data labels and low data quality. Transfer learning allows knowledge from one domain (the source domain) to be transferred to another domain (the target domain) to achieve better training results, which is suitable for this circumstance. Thus, Federated Transfer Learning

Table 2 An application sample of horizontal federated learning

Participants	Feature Record	Age	Married	Job	Salary	Credit	Loan
Bank 1	Client_11	30	Yes	Nurse	30k	No	NA
	Client_1n	50	No	Lecture	50k	Yes	NA
Bank 2	Client_21	40	No	Doctor	80k	Yes	No
	Client_2n	29	Yes	Worker	20k	No	Yes
Bank 3	Client_3n	37	Yes	Singer	100k	NA	Yes

Table 3 An application sample of vertical federated learning

Participants	Feature Record	Age	Married	Job	Bill	Item	Time
Bank X	Client_A	30	Yes	Nurse	NA	NA	NA
	Client_B	50	No	Lecture	NA	NA	NA
Retail Y	Client_A	NA	NA	NA	80k	Fish	10 A.M.
	Client_C	NA	NA	NA	40k	Meat	4 P.M.
	Client_D	NA	NA	NA	60k	Beef	5 P.M.

Table 4 Credit card dataset

Total transaction	Fraud	Normal	Label fraud	Label normal
284,807	492	284,315	1	0

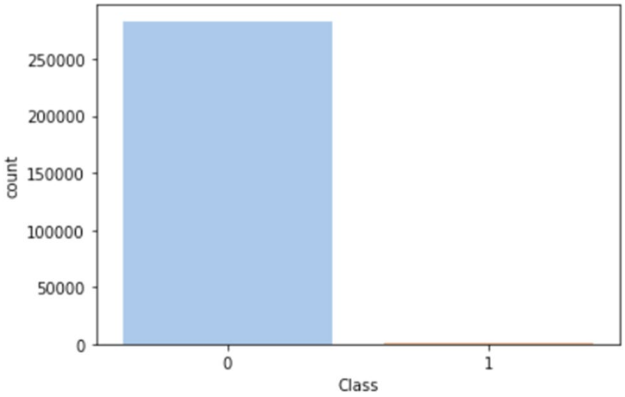


Fig. 3 Unbalanced credit card fraud dataset visualization

Table 6 The parameters for the federated learning experimentation

Parameter	Value
Training round	50, 75, 100, 125, 150
Epoch	25, 50, 75, 100
Learning rate	0.01, 0.02, 0.03, 0.04, 0.05
Batch size	128, 256, 512, 1024, 2048, 4096

(FTL) was conceived [34] to generalize FL for broader applications, especially when dealing with common parties with little overlap, as depicted in Fig. 4. This represents the first complete stack for FL based on transfer learning, encompassing training, evaluation, and cross-validation. Besides, the neural networks with additive homomorphic encryption mechanics cannot only prevent privacy leakage but also provide comparable accuracy to traditional non-privacy-preserving models.

The Experimentation Results

Dataset Description

In this review work, we conducted a series of comprehensive experiments to manifest the superiority of our method. The experiment dataset from the European Credit Card (ECC) transactions were made in September 2013 by European cardholders and it was provided by the ULB ML Group [25]. This dataset contained total of 284,807 anonymized transactions spanning over a period of two days, but there were only 492 fraudulent transactions in this dataset with a ratio of 1:578. The dataset was highly imbalanced as it had been observed only 0.172% of the transactions that were fraudulent. Due to confidentiality issues, the original features, and some background information about the dataset could not be provided. It was described in Table 4 before preprocessing data was performed to drop unwanted data (null value) and filter out necessary data. This was a classic example of an unbalanced dataset of credit card fraud (Fig. 3), it was very necessary to rebalance the raw data to prevent the classifiers from overfitting the legitimate class and ignore the patterns of frauds.

The Experimentation

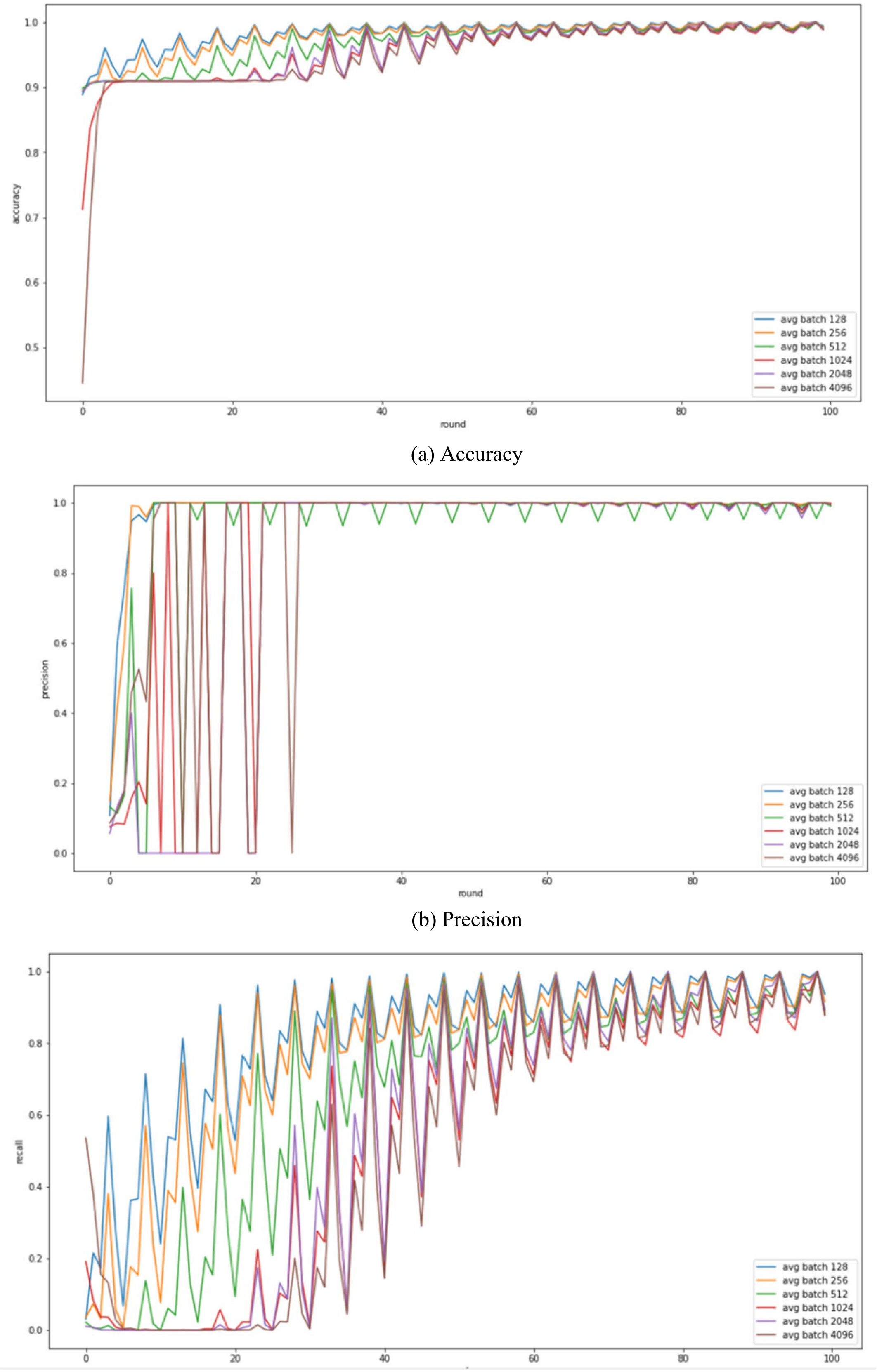
The first stage of the experiment involved preprocessing to clean the dataset. Each subject in the dataset was then treated as a client, resulting in four different clients. To minimize the risk of overfitting, each dataset was divided into 80% training data and 20% testing data. Subsequently, the shared global model on the central server was a multi-layer perceptron consisting of four layers. The first layer served as the input layer, comprising 29 input neurons receiving features from the dataset. This was followed by two hidden layers: the first with 128 nodes and the second with 64 nodes, with each layer utilizing a rectified linear unit as the activation function. The final layer consisted of a SoftMax output layer with sigmoid activation, as depicted in Table 5.

For the federated learning experimentation, a series of experiments were conducted on various parameters to optimize the model, including learning rate, batch size, epochs, and training rounds, as depicted in Table 6 for federated averaging (FedAvg). The models were developed using TensorFlow with GPU for full implementations.

In the experimental setup, we configured four clients and one central server, utilizing the vertical federated learning model for the training process.

Table 5 A fully connected multi-layer neural network

Layer	Node number	Activation function	Loss function	Learning rate
Input	29		Binary cross entropy	0.01–0.05
Hidden (1)	128	ReLu	Binary cross entropy	0.01–0.05
Hidden (2)	64	ReLu	Binary cross entropy	0.01–0.05
Output	1	Sigmoid	Binary cross entropy	0.01–0.05



(a) Accuracy

(b) Precision

(c) Recall

Fig. 4 The performance measure of local batch size

Table 7 The results for each training round

Round	Precision	Recall
50	0.8666667	0.61904764
75	0.86440676	0.8095238
100	0.8730159	0.8730159
125	**0.875**	**0.8888889**
150	0.67469877	0.8888889

Bold value shows enable readers to discern that Round 125 yielded superior results compared to the other rounds

1. The server randomly selects three clients from the four available.
2. Each selected client uses the fully connected multi-layer neural network as shown in Table 5 and begins training it on their local dataset.
3. After completing a specified number of training passes, each client sends their locally trained model back to the server.
4. The server then averages the model parameters from the three clients to produce the latest aggregate model parameters.
5. Repeat step 1 until the preset number of iterations.

Results and Discussions

Figure 4 demonstrated the impact of the performance measure for credit card fraud detection system, and the batch size with the training round influenced accuracy, recall, and precision. When the batch size was increased from 128 to 4096, the model appeared to achieve faster convergence and yield better results. Table 7 confirmed that a batch size of 256 achieved the best training stability and generalization performance. For precision parameters, there was a stable value from 50 to 125 training rounds, but it decreased sharply after surpassing 125 rounds, whereas recall parameters increased gradually when the number of rounds increased and converged at 125 rounds. This was the problem when detecting fraudulent credit card transactions, which was called an imbalance dataset. The raw imbalanced dataset was one in which the number of data points for fraudulent class was significantly lower than the number of data points for normal class. This could lead to overfit for precision parameters. Fortunately, there was a widely used technique for dealing with class imbalance that could help overcome these issues, which was called SMOTE. This method addresses data imbalance by oversampling minority instances (fraud cases) using nearest neighbors of fraud cases to create new synthetic fraud cases.

Table 8 showed that the federated FDS with a data balancing mechanism outperforms those trained with raw data. The improved fraud detection system performed better with a higher proportion of fraudulent transactions. This is because the FDS could better learn patterns of both fraud and legitimate transactions when the data was more balanced. Achieving a 1:1 ratio through sampling yielded the best result.

Conclusion and Future Work

This study proposes a federated approach to fraud detection to address fraud detection challenges and build an effective fraud detection system (FDS) in data-scarce environments. The research contributions include implementing an FDS in a vertical federated learning environment and evaluating the federated learning model using real-world datasets from European cardholders. This paper builds an FDS that has allowed the cooperative banks to reap the benefits of the shared model, which has detected more fraud than individual banks without sharing data together. Thus, the cardholder's sensitive information is protected. Federated learning algorithms can be applied to combat credit card fraud detection. This decentralized data method can protect dataset sensitivity and security, reducing the impact of unavailable datasets to some extent. However, privacy issues still exist in the federated learning fraud detection system. Firstly, we need to consider what sensitive information can be revealed by the global parameters [35]. For another consideration, we should reflect on what sensitive data can be learned by gaining access to parameter updates of an individual bank. In federated learning, local updates and global model parameters are leaked in "honest but curious" and untrusted execution environments. In future endeavors, the FDS is implemented using Federated Transfer Learning. This learning scheme

Table 8 The result for the ratio between normal and fraudulent

The ratio between normal and fraudulent	Precision	Recall
Raw dataset	0.8358209	0.8888889
The reduction of 90% of fraudulent transactions	0.8730159	0.8730159
The reduction of 80% of fraudulent transactions	0.86885244	0.84126985
The reduction of 70% of fraudulent transactions	0.8666667	0.82539684
The reduction of 60% of fraudulent transactions	0.86885244	0.84126985
The reduction of 50% of fraudulent transactions	0.86885244	0.84126985

involves transferring knowledge from parties with rich feature spaces to those without sufficient features or from multiple sources, while still preserving privacy.

Acknowledgements This research is funded by University of Information Technology-Vietnam National University Ho Chi Minh City under grant number D1-2023-19.

Data availability The data that support the findings of this study are available from Kaggle at https://www.kaggle.com/datasets/mlgulb/creditcardfraud. Due to privacy or ethical restrictions, certain data are available upon reasonable request from the corresponding author.

Declarations

Conflict of Interest The authors report no conflicts of interest.

References

1. Bahnsen AC, Aouada D, Stojanovic A, Ottersten B. Feature engineering strategies for credit card fraud detection. Expert Syst Appl. 2016;51:134–42.
2. Zareapoor M, Shamsolmoali P. Application of credit card fraud detection: based on bagging ensemble classifier. Procedia Comput Sci. 2015;48(2015):679–85.
3. Bolton RJ, Hand DJ. Statistical fraud detection: a review. Stat Sci. 2002;17(3):235–55.
4. Sahin Y, Bulkan S, Duman E. A cost-sensitive decision tree approach for fraud detection. Expert Syst Appl. 2013;40(15):5916–23.
5. Laleh N, Abdollahi Azgomi M. A taxonomy of frauds and fraud detection techniques. In: Information systems, technology and management: third international conference. 2009. p. 256–67.
6. Delamaire L, Abdou H, Pointon J. Credit card fraud and detection techniques: a review. Banks Bank Syst. 2009;4(2):57–68.
7. Srivastava A, Kundu A, Sural S, Majumdar A. Credit card fraud detection using hidden Markov model. IEEE Trans Dependable Secure Comput. 2008;5(1):37–48.
8. Abdallah A, Maarof MA, Zainal A. Fraud detection system: a survey. J Netw Comput Appl. 2016;68:90–113.
9. Ahmed M, Ansar K, Muckley CB, Khan A, Anjum A, Talha M. A semantic rule based digital fraud detection. PeerJ Comput Sci. 2021;7: e649.
10. Sherly KK, Nedunchezhian R. BOAT adaptive credit card fraud detection system. In: IEEE international conference on computational intelligence and computing research. 2010. p. 1–7.
11. Shemar AK, Sidhu BK. Credit card fraud detection using anomaly detection. J Innov Comput Sci Eng. 2020;10(1):7–12.
12. Bahnsen AC, Stojanovic A, Aouada D, Ottersten B. Improving credit card fraud detection with calibrated probabilities. In: Proceedings of the SIAM international conference on data mining. 2014. p. 677–85.
13. Tingfei H, Guangquan C, Kuihua H. Using variational auto encoding in credit card fraud detection. IEEE Access. 2020;8:149841–53.
14. Van Vlasselaer V, Bravo C, Caelen O, Eliassi-Rad T, Akoglu L, Snoeck M, Baesens B. APATE: a novel approach for automated credit card transaction fraud detection using network-based extensions. Decis Support Syst. 2015;75:38–48.
15. Soemers D, Brys T, Driessens K, Winands M, Nowé A. Adapting to concept drift in credit card transaction data streams using contextual bandits and decision trees. In: Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1. 2018.
16. Chen RC, Chen TS, Lin CC. A new binary support vector system for increasing detection rate of credit card fraud. Int J Pattern Recogn Artif Intell. 2006;20(02):227–39.
17. Dang TK, Tran TC, Tuan LM, Tiep MV. Machine learning based on resampling approaches and deep reinforcement learning for credit card fraud detection systems. Appl Sci. 2021;11(21):10004.
18. Dal Pozzolo A, Caelen O, Le Borgne YA, Waterschoot S, Bontempi G. Learned lessons in credit card fraud detection from a practitioner perspective. Expert Syst Appl. 2014;41(10):4915–28.
19. Tran TC, Dang TK. Machine learning for prediction of imbalanced data: credit fraud detection. In: 15th international conference on ubiquitous information management and communication (IMCOM). 2021. p. 1–7.
20. Taha AA, Malebary SJ. An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE Access. 2020;8:25579–87.
21. Jemima Jebaseeli T, Venkatesan R, Ramalakshmi K. Fraud detection for credit card transactions using random forest algorithm. In: Intelligence in big data technologies—beyond the hype: proceedings of ICBDCC 2019. 2021. p. 189–97.
22. Bolton RJ, Hand DJ. Unsupervised profiling methods for fraud detection. Credit scoring and credit control VII. 2001. p. 235–55.
23. Carneiro EM, Dias LAV, Da Cunha AM, Mialaret LFS. Cluster analysis and artificial neural networks: a case study in credit card fraud detection. In: 12th international conference on information technology-new generations. 2015. p. 122–6.
24. Asha RB, Suresh Kumar KR. Credit card fraud detection using artificial neural network. Glob Transitions Proc. 2021;2(1):35–41.
25. Misra S, Thakur S, Ghosh M, Saha SK. An autoencoder based model for detecting fraudulent credit card transaction. Procedia Comput Sci. 2020;167:254–62.
26. Sahin Y, Duman E. Detecting credit card fraud by ANN and logistic regression. In: International symposium on innovations in intelligent systems and applications. 2011. p. 315–9.
27. Dang TK, Tran-Truong PT, Trang NTH. An enhanced incentive mechanism for crowdsourced federated learning based on contract theory and shapley value. In: The 10th international conference on future data and security engineering. 2023. p. 18–33.
28. Pham HT, Nguyen KN, Phun VH, Dang TK. Secure recommender system based on neural collaborative filtering and federated learning. In: The 16th international conference on advanced computing and analytics (ACOMPA). 2022. p. 1–11.
29. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. 2017. p. 1273–82.
30. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol. 2019;10(2):1–19.
31. West J, Bhattacharya M. Some experimental issues in financial fraud mining. Procedia Comput Sci. 2016;80:1734–44.
32. Gascón A, Schoppmann P, Balle B, Raykova M, Doerner J, Zahur S, Evans D. Secure linear regression on vertically partitioned datasets. IACR Cryptol. ePrint Arch. 2016. p. 892.
33. Xu R, Baracaldo N, Zhou Y, Anwar A, Joshi J, Ludwig H. Fedv: privacy-preserving federated learning over vertically partitioned data. In: Proceedings of the 14th ACM workshop on artificial intelligence and security. 2021. p. 181–92.
34. Liu Y, Yang Q, Chen T. Federated learning and transfer learning for privacy, security and confidentiality. In: The 33rd AAAI conference on artificial intelligence (AAAI). 2019.

35. Ha T, Dang TK, Le H, Truong TA. Security and privacy issues in deep learning: a brief review. SN Comput Sci. 2020;1(5):253.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.