Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **9**, Number **4**, November **2024**

Financial Fraud Recognition Model Based on Privacy Preserving and Federated Learning

Qiu-Xiu Liu

School of Business
Yulin Normal University, Yulin 537000, P. R. China
jan19912024@163.com

Jian-Guo Zhang*

School of Business
Yulin Normal University, Yulin 537000, P. R. China
royzhang1107@163.com

Boris Tan

St. Paul University Philippines, Tuguegarao 3500, Philippines
jt0084@163.com

*Corresponding author: Jian-Guo Zhang
Received January 17, 2024, revised April 7, 2024, accepted July 22, 2024.

Abstract. *In financial fraud identification tasks, it is often necessary to integrate data from different institutions or organisations, but these data often contain sensitive information, so achieving data privacy and data sharing at the same time is a huge challenge. Privacy-preserving machine learning techniques can make it possible for all parties to train models together without revealing sensitive information by using techniques such as encryption and differential privacy. Therefore, this work proposes the use of federated learning techniques in distributed machine learning approaches for modelling and completing the classification and identification of financial fraud. Firstly, the definition of financial fraud is given and the principle of federation learning in privacy-preserving machine learning is analysed. Then, the federated meta-learning FedMeta framework is used to construct the financial fraud identification model in order to better solve the financial data heterogeneity problem and the temperature drift problem. During the metamodel training process, the gradient is cut before uploading the local gradient in each round to determine the privacy sensitivity of the gradient. Differential privacy noise protection is applied to the gradient, and the perturbed gradient information is uploaded to the server side, which avoids the original gradient from leaking the user's privacy information. Secondly, for the problem that gradient encryption with single-key homomorphism cannot resist conspiracy attacks, Paillier homomorphic encryption is used to encrypt the user gradient parameters. The experimental results show that the proposed model performs well in all the metrics, and has better overall performance in the financial fraud identification task compared to SecureBoost and longitudinal neural network models.*
Keywords: Financial fraud; Privacy-preserving machine learning; Differential privacy; Federated learning; Meta-learning; Homomorphic encryption

1. **Introduction.** Enterprises are microscopic subjects of market operation, and their disclosed operating conditions provide important data support for economic development. And financial fraud will interfere with the normal business order and impact on the whole economy and society. Therefore, identifying financial fraud has become a classic topic

[1, 2]. At the same time, the use of machine learning to identify financial fraud has also become a popular research method [3, 4, 5].

Privacy-preserving machine learning techniques provide a way to solve the above problems. Privacy-preserving machine learning is characterised by "data not moving model moving" [6, 7], i.e., the data is still in the hands of the data owners, and the joint modelling of multiple parties is achieved by sharing the model parameters without disclosing the data, and ultimately, machine learning is achieved by using all the data. This approach technically bypasses the difficulties of data trading, is less difficult to implement, and is more easily accepted by all parties, and thus has gradually gained popularity in settings where privacy protection and data confidentiality need to be emphasised [8].

Privacy-preserving machine learning techniques encompass three two technological routes [9, 10]: secure multi-party computation (arithmetic-logic operations using multi-party data); federated learning (machine learning model construction using multi-party data) and hardware encryption. Ideas and algorithms for secure multiparty computation emerged in the 1980s, and federated learning saw a wave of research following the release of Google's FedAvg algorithm in 2016 [11]. As the price of being able to prevent data leakage, the various methods mentioned above usually face the problems of complex computational steps and restrictive application conditions, and need to trade-off generality, computational speed, and reliability, so improved algorithms continue to appear, while specific problems need to be analysed in different scenarios. Privacy-preserving machine learning scenarios usually include financial, medical, personal sensitive data use, etc., in the financial field, mainly including insurance fraud identification, credit card fraud identification and other scenarios, and financial fraud identification and its similarity, so this paper plans to use the use of privacy-preserving machine learning technology for modelling to help financial fraud identification model application landing.

1.1. **Related Work.** In the motivation of financial fraud, with the continuous development of the capital market. The research on the causes of fraud is also deepening. The mainstream research is divided into iceberg theory [12], fraud triangle [13], GONE [14], fraud risk factor theory [15] and so on. In terms of financial fraud identification, there have been a large number of studies on fraud identification using machine learning models.

In financial fraud identification research, models mostly follow a simple to complex trend. Most of the studies use structured data (e.g., information on turnover rate, shareholding of major shareholders, etc.), and evolve from early single-layer models such as SVM, logistic regression, decision trees, etc., to complex models such as NN. Pai et al. [16] used Support Vector Machine (SVM) algorithm to classify financial statement data and predict whether a company has financial fraud or not. It extracts a large number of financial indicators as features, such as gearing ratio, cash flow, etc., and then trains and tests them with SVM classifiers. Shih et al. [17] used logistic regression algorithm to study publicly available data on corporate fraud cases. It extracted basic company information, financial data and management team data as features, and built a classification model for fraud and non-fraud prediction by logistic regression. The results show that the model has high precision and recall. Chen [18] used the CART decision tree algorithm to model the financial indicators as the node classification criteria, and obtained an interpretable decision tree model to detect financial fraud. Wu and Du [19] proposed a deep learning method based on LSTM for financial fraud prediction. It trained an LSTM hybrid network model using textual and numerical information from financial statements. The results show that this deep learning model outperforms traditional machine learning algorithms on this task.

In addition to the aforementioned research on the mechanisms of data trading itself, some research aims to clear the way for data trading through technological means. Privacy-preserving machine learning is one possible approach. Using cryptography, privacy-preserving machine learning is able to combine data from multiple parties for computation without disclosing the data, effectively solving the problem of moral hazard in data transactions. Research on such problems is usually divided into algorithm design and application research aspects. At the level of algorithmic research, Al-Rubaie and Chang [20] reviewed the relevant papers on privacy-preserving machine learning and summarised the research routes of machine learning, including secure multi-party computation and federated computation. Although privacy-preserving machine learning can protect the data privacy of the participants, the cost is a rise in computational complexity, coupled with the need to communicate among multiple participants, resulting in a decrease in computational efficiency. To address this problem, Park et al. [21] proposed a federated learning algorithm that optimises the communication cost by 90% over traditional algorithms such as FedAvg. You et al. [22] et al. designed a weight-based asynchronous federation learning aggregation update method, and tests on MNIST, CIFAR-10 showed that it can effectively improve the training efficiency of federation learning.

1.2. **Motivation and contribution.** Recent studies have found that although the original data is not exposed in federated learning, the model parameters exchanged can also result in the disclosure of some of the user's data privacy. In addition, in the data heterogeneity problem, different companies and organisations have different accounting and financial systems, and there is a great deal of heterogeneity in their financial data, leading to the inability of commonly used federated learning algorithms to perform poorly on some users. Therefore, in order to solve the above problem, it is proposed to use the federated meta-learning technique [23] in federated learning methods for modelling and completing the classification and identification of financial frauds. The main innovations and contributions of this work include:

(1) The FedMeta framework, a federated meta-learning framework, was chosen to build a financial fraud identification model for the financial data heterogeneity problem and the temperature drift problem, with the goal of collaboratively training the meta-model using data distributed among multiple clients.

(2) Since the training of federated meta-learning is also based on stochastic gradient descent, the same privacy leakage problem exists. In order to solve the above problems, a federated meta-learning method based on differential privacy preservation is designed to protect the user's privacy security, which is more suitable for financial fraud recognition models.

(3) Aiming at the problem that the stochastic gradient descent algorithm with single-key homomorphism is unable to resist the conspiracy attack, it is proposed to use Paillier homomorphic encryption to encrypt the user's local model parameters.

2. Concepts of Financial Fraud and Privacy-Preserving Machine Learning Theory.

2.1. **Definition of financial fraud.** Financial fraud refers to intentional misstatements related to the audit of financial statements, including misstatements resulting from the preparation of false financial reports and misstatements resulting from misappropriation of assets. Therefore, with reference to the above provisions, this paper considers financial fraud as the act of falsifying, altering, or providing false accounting documents, books, and reports. This paper uses the library of penalty documents obtained from the China Stock Market & Accounting Research (CSMAR) database to identify fraudulent firms.

Iceberg theory is one of the early studies on the motivation of financial fraud. The theory divides the motivation of financial fraud into two parts, one of which can be identified by observing written institutional documents and tracking the normal business and management processes of an enterprise; the other part is the psychological activities, emotions, values and other factors of the parties involved, which are usually difficult to be quantified and identified and confirmed by external evidence.

The fraud triangle theory attributes the occurrence of financial fraud to three factors: pressure, opportunity and excuse. After the Fraud Triangle and GNOE, various studies have continued to expand the factors influencing financial fraud. The Risk Factor Theory, which incorporates these factors, constitutes one of the most well-developed theories of financial fraud. The theory categorises financial fraud factors into individual risk factors and general risk factors. Individual risk factors are factors controlled by different individuals, which usually vary from person to person and are not controlled by the enterprise, including personal values, moral level, etc. General risk factors are controlled by the enterprise, such as the chances of fraud, the probability of being detected, and the penalties after being detected.

2.2. **Privacy-preserving machine learning.** "Privacy" refers to any data that the data owner does not want to disclose or put at the disposal of other subjects. There are a number of semi-honest data owners (who do not provide false data but inquire about the data of other parties) who combine their data for machine learning training to solve a machine learning problem, and these data owners do not want to disclose or share the data with other owners. The techniques that enable machine learning at this point are called privacy-preserving machine learning techniques. The use of Internet information can improve the effectiveness of financial fraud identification. When Internet platforms do not want to disclose their data directly to other subjects, this Internet information can be called "private".

In privacy-preserving machine learning, sometimes the parties involved are not equal: one party owns the sample labels and some of the sample features, and actively obtains more data from the other party for modelling purposes, e.g., an investor organisation knows whether a company has committed financial fraud and wants to obtain information from other parties (e.g., an internet platform) for modelling purposes. The other party only provides data services to the outside world, e.g., the Internet platform does not care whether a company is financially fraudulent, but only provides data at the request of other parties (e.g., investors). The former is called the active party and the latter is called the passive party. Usually, the active party initiates the modelling request and manages the model structure, parameters, and states, while the passive party calculates the corresponding parameters upon the request of the active party.

Privacy-preserving machine learning techniques are classified into three main technical routes. (1) Secure multiparty computation, with the goal of implementing arithmetic, comparative, and logical operations using multiparty data without disclosing the data of both parties, encompassing homomorphic encryption, obfuscated circuits, and secret sharing. (2) Federated learning with the goal of implementing machine learning models [24, 25]. Each data owner agrees to use the same model structure and computes weights using their own data, and subsequently aggregates the weights to train the model. (3) Hardware encryption, in which hardware such as trusted platform modules are used to ensure that data is not compromised. This work mainly deals with the first two techniques as hardware devices are more expensive.

2.3. **Principles of federated learning.** Federated learning is a new approach to machine learning that is able to use sensitive datasets distributed across a large number of

clients (these may be mobile phones, other mobile devices, or sensors) without having to collect data from them. In federated learning, users upload their local model parameters in each iteration to train a global model. In order to update the global model, a centralised server will be used to aggregate the parameters received from these users and send the updated global model, back to them. Thanks to its ability to avoid users from leaking private data to other participants, federated learning has become one of the fundamental techniques for security-sensitive tasks.

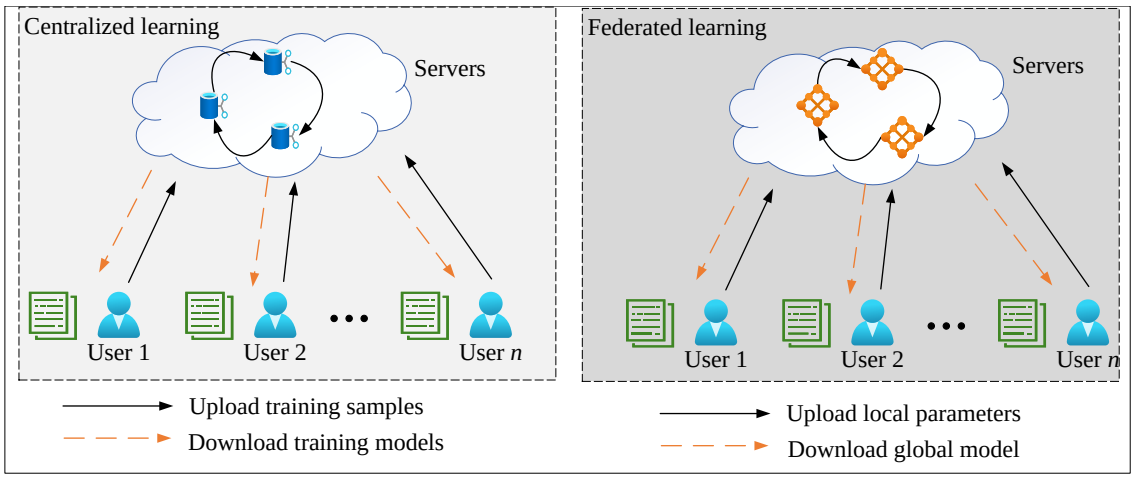

Figure 1. Generalised machine learning training frameworks

According to different training methods, machine learning can be divided into two types: centralised learning and federated learning, as shown in Figure 1. Federated learning mainly consists of multiple users and a cloud server, where users first train a local model based on the processed raw data, and then send part of the training results, i.e., the model parameters (e.g., training gradient), to the server, which aggregates all the parameters and then resends the results to each user. Unlike traditional centralised machine learning, federated learning does not need to collect raw data from all users for learning and training during the training process, which can greatly reduce the privacy risk and cost of the system, thus breaking the barrier of data silos. During the training process of federated learning, the data owner uses a gradient descent algorithm to minimise the loss function in order to find the optimal parameters. The process is described as follows [26]:

(1) In the t-th round of training, each data owner U_i trains the local model θ_i based on the local private dataset D_i, and computes the local gradient ∇g_i^t.

$$\nabla g_i^t = \frac{\partial F\left(y_i, f\left(x_i\right)\right)}{\partial \theta^t} \tag{1}$$

where $F_i(\theta)$ is the loss function of the data owner U_i based on the data set D_i.

(2) All data owners U_i upload the gradient ∇g_i^t to the FL server.

(3) After the FL server collects the gradients $G' = \{\nabla g_1^t, \nabla g_2^t, \ldots, \nabla g_m^t\}$ sent by the m users, it performs additive aggregation.

$$g_{\text{global}}^t = \frac{1}{m} \sum_{i=1}^{m} \nabla g_i^t \tag{2}$$

(4) The federated learning server sends the aggregated gradients back to all participants [28], and the users make the corresponding model updates.

$$\theta_i^{t+1} = \theta_i^t - lr * \frac{g_{\text{global}}^t}{m} \tag{3}$$

where lr is the learning rate, i.e., the step size to move in the opposite direction of the gradient.

The above training process of federated learning is iterated until the loss function $F_i(\theta)$ reaches a certain threshold or the accuracy of the local model has been reached.

3. **Privacy-preserving and federated learning-based model for financial fraud identification.**

3.1. **Model construction based on federated meta-learning.**

When constructing financial fraud identification models, due to data ownership, privacy regulations, and other issues, companies are often reluctant to share financial data, so the federated learning framework is highly adaptable. Currently, there are a number of commonly used federated learning algorithms, such as SecureBoost [27] and Longitudinal Neural Network (LNN). However, federated meta-learning (FedMeta) [29] is more suitable for building financial fraud identification models than SecureBoost and longitudinal neural networks, mainly because of the financial data heterogeneity problem and the temperature drift problem. In the data heterogeneity problem, different companies and organizations have different accounting and financial systems and their financial data are highly heterogeneous. In the case of the FedMeta framework, each client performs personalized training of the local model to be able to better adapt to the respective data distribution, and then aggregates and updates the parameter gradients globally, which can cover a greater diversity of data.

In the temperature drift problem, financial fraud behavior is related to changes in a variety of factors, such as the characteristics of the financial data itself (e.g., cyclicality, seasonality), the macroeconomic environment, and the corporate governance structure, and there is a certain degree of temporal drift. Federated meta-learning allows each node to train and update the model locally, which can capture and respond to changes in data distribution in real time.

Therefore, the FedMeta framework is chosen in this work to build a financial fraud recognition model. The goal of meta-learning is to train a model, such as a deep neural network, in multiple tasks that can be better applied to new tasks by fast adaptation with samples on new tasks. The more classical MAML model in meta-learning uses a gradient-based learning rule, which allows for fast adaptation on new tasks.

The goal of meta-learning is to learn to generate sensitive model parameters θ that, when encountering a new task, produce an effective improvement in the model's loss function for the new task t when the model parameters are changed slightly in the direction of the gradient, as shown in Figure 2.

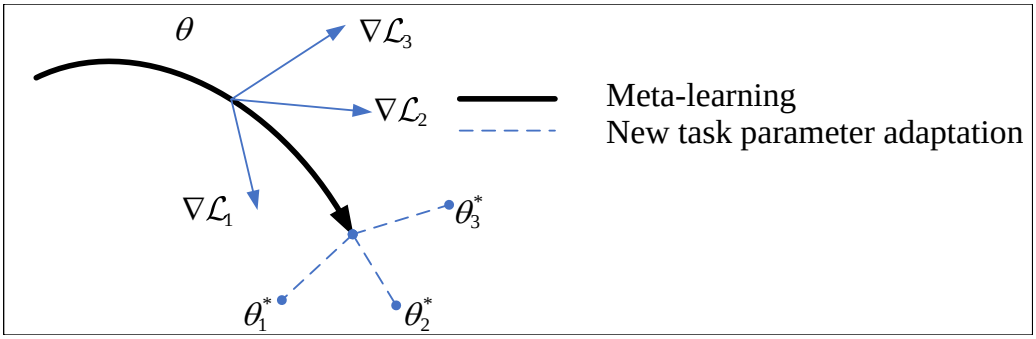

Figure 2. MAML parameter adaptation process

The parameter vector θ is updated by performing a stochastic gradient descent on the task T_i to obtain the updated vector θ_i'. For example, when using a gradient update, the update is done as follows:

$$\theta_i' = \theta - \alpha \nabla_\theta L_{T_i}(f_\theta) \tag{4}$$

where the learning rate α is a fixed hyperparameter. More specifically, the goal of meta-learning is to find the meta-model parameter θ that minimises the overall loss of the parameter when adapted to a new task, and the objective function of meta-learning is as follows.

$$\min_\theta \sum_{T_i \sim p(T)} L_{T_i}\left(f_{\theta_i'}\right) = \sum_{T_i \sim p(T)} L_{T_i}\left(f_\theta - \alpha \nabla_\theta L_{T_i}(f_\theta)\right) \tag{5}$$

Meta-optimisation is performed on the model parameter θ and the objective is computed using the updated model parameter θ_i'. For cross-task meta-optimisation by Stochastic Gradient Descent (SGD), the model parameters θ are updated as follows.

$$\theta = \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} L_{T_i}\left(f_{\theta_i'}\right) \tag{6}$$

where β is the learning rate for updating the metamodel parameters.

FedMeta, a federated meta-learning framework, is meta-learning implemented in a federated setting with the goal of collaboratively training meta-models using data distributed across multiple clients. Correspond each client in federated learning to each task in meta-learning. The server will perform a weighted average of the local parameters based on the number of samples in the clients. The aggregated model parameters are then redistributed to different clients as models for the next round of updates.

$$\theta_k = \theta - \eta g_k \tag{7}$$

Where k represents the client, η is the learning rate of the client's gradient update, g_k is the gradient of the client k, θ_k denotes the updated parameters of the k-th client, and θ denotes the model parameters assigned by the server to the client.

The update is performed on the server side as follows.

$$\theta = \sum_{k=1}^{K} \frac{n_k}{n} \theta_k \tag{8}$$

where n_k is the number of samples for the k-th client and n is the total number of samples for all clients. Each client can iterate multiple times before returning a local model update in order to update the model parameters assigned by the server.

3.2. **Differential privacy preserving federated meta-learning.** In previous federated learning research it was argued that gradient sharing in federated learning does not expose information about a user's training data. However, some recent studies have shown that gradients can expose training data to some extent. The privacy disclosure issue during federated learning is shown in Figure 3. When performing federation training, the parameter server is able to steal local information from the gradients returned from client nodes. In each training round t, each client node i receives the model parameters W from the server and samples a batch (x_t^i, y_t^i) in its local dataset to compute the gradient.

$$\nabla W_i = \frac{\partial\left(F\left(x_t^i, W\right), y_t^i\right)}{\partial W} \tag{9}$$

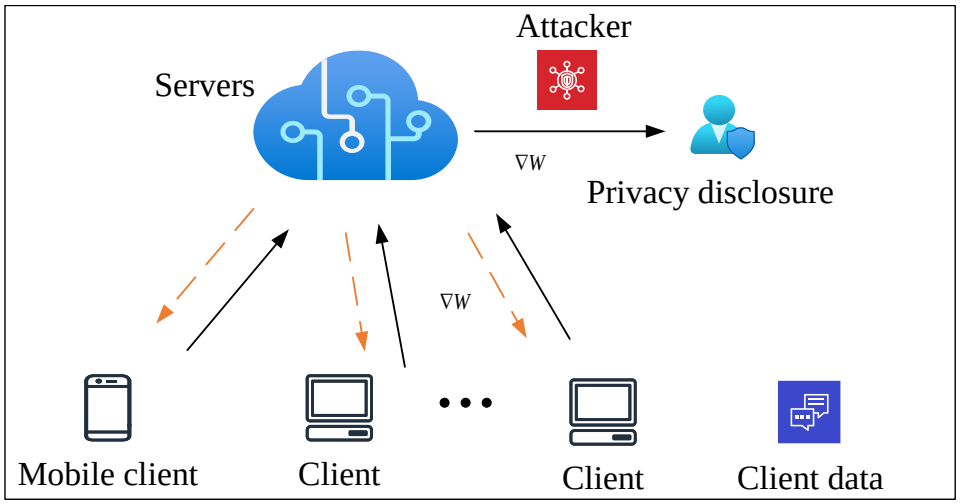

Figure 3. Privacy Breaches in Federal Learning

The local gradients of each client are averaged over the parameter server ∇W, and then the parameters of the federated model are updated.

$$\nabla W = \frac{1}{N}\sum_{1}^{N}\nabla W_i; W = W - \eta\nabla W \tag{10}$$

where N is the number of clients, η is the server-side learning rate, and ∇W_i is the parameter uploaded by client i.

The server tries to reason about the client's local data while completing normal training. For receiving a gradient $\nabla W_{i,t,k}$ from client i, the goal is to steal the training data (x_t^i, y_t^i) of client user i. In order to recover the user's original data from the gradient, a dummy input x_t' and an input label y_t' are randomly initialised. Then, these "dummy data" are fed into the model and a "dummy gradient" is obtained.

$$\nabla W' = \frac{\partial\left(F\left(x_t', W\right), y'\right)}{\partial W} \tag{11}$$

Through multiple rounds of training, the loss function is minimised to make the virtual gradient closer to the real gradient, thus making the virtual input close to the real training data. It can be seen that in some distributed federated learning scenarios, even though the original data is kept locally all the time, the gradient based on the exchange can still cause privacy leakage to the user. Since the training of federated meta-learning is also based on stochastic gradient descent, the same privacy leakage problem exists. In order to solve the above problems, this paper designs a federated meta-learning method based on differential privacy preservation, which protects the user's privacy security and is more applicable to financial fraud recognition models.

Based on FedMeta, a federated meta-learning framework, this paper proposes a differential privacy-preserving federated meta-learning method, DPP-FedMeta.By utilising differential privacy-preserving privacy information of the participating training users, the method makes a trade-off between model utility and privacy preservation. Figure 4 shows the differential privacy-preserving federated meta-learning training process. Users are primarily responsible for the local model parameters trained on their private data, while preventing indirect leakage of private information to honest and curious parameter servers or other participants.

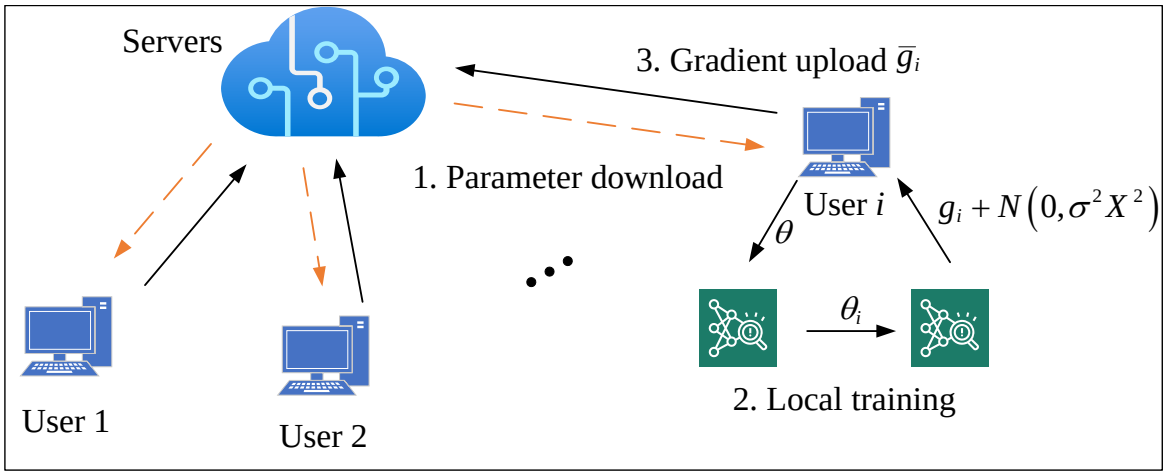

Figure 4. Differential Privacy Federated Meta-Learning Framework

The goal of federated meta-learning is to train a meta-learning model using clients C_i with congested data D_i, coordinated by a parameter server S. The end result of co-training here is not the final parameters of the model, but rather the initialised metamodel parameters of the model on each client. When this metamodel parameter is assigned to a client or a new user, it needs to be trained with gradient descent using the client's local data to quickly match the new task. Clients are coordinated by a central server to complete each round of training tasks.

The server first randomly initialises the meta-model parameters θ and sends the initialisation parameters to each client. After receiving the initialisation parameters from the server, the client first performs a local update of the parameters based on its own local support set. This allows the federated metamodel to do fast adaptation on new users in the future, i.e., only a small amount of gradient descent is used to get better accuracy on new users. After getting the initialisation parameters θ, the client first calculates the loss values based on the local support set.

$$L_{D_s^i}(\theta) = \frac{1}{|D_s^i|} \sum_{(x,y)\in D_s^i} \ell\left(f_\theta(x), y\right) \tag{12}$$

where D_s^i is the user's support set.

Calculate the gradient based on the loss value of the initialised model parameters on the support set, and perform a gradient descent by taking a step in the opposite direction of this gradient.

$$\theta_i = \theta - \alpha \nabla_\theta L_{D_s^i}(\theta) \tag{13}$$

This operation updates the meta-model parameters to complete the local adaptation of the incoming model. The α is the learning rate of the client gradient update, and the updated model parameters are θ_i. Next, the loss function is computed on the local query set using the adapted metamodel.

$$L_{D_q^i}(\theta_i) = \frac{1}{|D_q^i|} \sum_{(x,y)\in D_q^i} \ell\left(f_{\theta_i}(x), y\right) \tag{14}$$

where D_q^i is the user's query set.

Calculate the user's gradient at this point.

$$g_i = \nabla L_{D_q^i}(\theta_i) \tag{15}$$

The client needs to send the trained gradient g_i to the server for global meta-model parameter update. It is assumed that in general, the server is honest and curious, i.e., the server does not break the training rules of the model, but explores the user's privacy information from the gradient uploaded by the client. Unlike the commonly used gradient privacy protection methods, this work utilises a differential privacy mechanism to perturb the local gradient uploaded by the client in each round with Gaussian noise.

The key to the federated meta-learning differential privacy training approach is to implement a differential privacy-preserving stochastic gradient descent (SGD) algorithm. Instead of exposing the entire local gradient to the server side after the client obtains the local gradient, a differential privacy preserving mechanism is used to process the original gradient. A common cutting method of SGD in deep learning is used to avoid gradient explosion. The method of cutting the gradient for each sample is shown as follows.

$$\overline{g_i}(x_q) = \frac{g_i(x_q)}{\max\left(1, \frac{\|g_i(x_q)\|_2}{X}\right)} \tag{16}$$

For the cut gradient, a differential privacy Gaussian noise perturbation is added.

$$\overline{g_i} = \frac{1}{|D_q^i|}\left(\sum_q \overline{g_i}(x_q) + N(0, \sigma^2 X^2)\right) \tag{17}$$

The local gradient $\overline{g_i}$ of the user with noise is obtained, and the client sends the local gradient after this round of perturbation to the server for global model update.

After receiving the noise gradient returned from each client, the server side averages the noise gradient and performs gradient descent in the direction opposite to the gradient to update the initialised meta-model maintained on the server side.

$$\theta = \theta - \frac{\beta}{|C|}\sum_{i \in C} \overline{g_i} \tag{18}$$

Where β is the learning rate of the gradient update performed by the server, and $|C|$ is the number of clients. The updated initialisation parameters are sent to the clients for the next round of training until the federated meta-model converges.

3.3. **Paillier-based SGD Gradient Encryption.** The above model uses the SGD algorithm to update the parameters of the local and global models during training. However, the single-key homomorphic SGD algorithm cannot resist the problem of conspiracy attacks. Therefore, in order to further ensure the privacy of user data, this paper uses Paillier homomorphic encryption to encrypt the user's local model parameters, and uses the additive homomorphic property to ensure that the FedMeta server is able to perform data aggregation in a dense state.

The data owners take the security parameters as input and jointly generate a pair of homomorphic keys (pka, ska) by running the key generation algorithm for Paillier homomorphic encryption. The public key (pka) is used to encrypt each data owner's local model parameters, which in the scheme of this chapter refer to the gradient parameters. The private key (ska) is used to decrypt the global gradient. The private key (ska) is kept between the data owners throughout the training process.

In each round of training, each data owner U_i participating in the training is based on the local dataset $D_i = \{(x_i, y_i)\}$, x_i denotes the input data, and y_i denotes the corresponding labels. Calculate the loss function $L_f(D_i, W_i)$. The smaller the value of the loss

function, the closer the predicted value is to the true value, i.e., the higher the accuracy of the model training.

$$L_f(D_i, W_i) = \frac{1}{|D_i|} \sum_{(x_i, y_i) \in D_i} (y_i - f(x_i, W_i))^2 \tag{19}$$

where W_i denotes the model parameters.

Then the data owner computes the gradient vector ∇g_i using a gradient descent algorithm on the loss function $L_f(D_i, W_i)$ computed in Equation (19).

$$\nabla g_i = \frac{\partial L_f(D_i, W_i)}{\partial \omega} \tag{20}$$

The Paillier homomorphic encryption algorithm is used to encrypt the participant's gradient ∇g_i to obtain the ciphertext gradient ∇c_i.

$$c_i = Enc_{pka}(\nabla g_i) \tag{21}$$

The use of the Paillier encryption algorithm, which satisfies the additive homomorphism property, not only serves to blind the gradient, but also facilitates the federated meta-server to perform model parameter aggregation in the dense state.

4. Experimental results and analyses.

4.1. Experimental environment.

The running hardware and software information for data acquisition and model construction in this paper is shown in Table 1. The model construction process sets up an assumption that the investor owns and uses the financial and non-financial data of the firm, owns the labels of whether the firm is fraudulent or not, and is the passive party in the privacy-preserving machine learning model.

Table 1. Model runtime platform hardware and software

Hardware/Software	Model/Version
CPU	Intel Xeon CPU E3-1265L v3 @ 2.50GHz 4-Core
RAM	32G DDR3 RDIMM
hard drive	256GSSD (system disc) + 1THDD x 3 (ZFS, data disc)
Hypervisor	VMware ESXi 6.7U3
Os	Ubuntu 18.0.5-LTS
Python	3.8
Scikit-learn	1.0.2
TensorFlow	2.3.4
FATE	1.8.0

4.2. Sample selection.

Before starting to obtain data, it is first necessary to determine the fraud and non-fraud samples. To ensure the objectivity and reliability of the results, the financial fraud samples are obtained from the Violation Information Summary Table of the Violation Incident Database in CSMAR, which records the violations of listed companies disclosed by the CSRC, local securities regulators, stock exchanges, and regulatory bureaus of various parts of the Ministry of Finance. Considering that the SEC updated its industry classification standards in 2012, the time period is set from 2012 to 2022 in order to provide sufficient time for the industry classification calibre and exposure to fraud. Considering that some of the samples may have financial malpractice that has not been proven (i.e., the "grey sample"), all non-financial malpractice-penalised firms in the table

are excluded from the overall population in this paper. Finally, a large number of samples with missing data are also excluded. After these filters and the treatment of missing values, a total of 17,231 samples are generated, of which 516 are fraud samples and 16,715 are non-fraud samples. Considering the large dimensionality of the model's features, it has been difficult to get a global view of the data using descriptive statistics. In this paper, we use t-SNE to downscale the data in order to have a global understanding of the data. t-SNE transforms the distance between each point in the high-dimensional space into a conditional probability that obeys a normal distribution. The distances between data points in the embedding space are made to obey a t-distribution, and the KL scatter, which measures the distribution in the high-dimensional space and the distribution in the embedding space, is used as the cost function.

4.3. **Performance evaluation and comparison.** For the machine learning task, the way to assess the effectiveness of the model is to perform inference using the test set and compare the results of its inference in the test set with the real labels in order to evaluate how effective the model is. Since financial fraud identification is a classification problem, it will be evaluated using metrics for classification problems. The vast majority of the metrics are based on the confusion matrix. The confusion matrix indicates the matrix of true labels and model predictions for all samples. The matrix is divided into four parts as shown in Table 2.

Table 2. Schematic Confusion Matrix

	Projected value	
real value	**0-Non-fraud**	**1-Fraud**
0-Non-fraud	TN - True Negative Example	FP - False Positive Example
1-Fraud	FN - False Negative Example	TP - True Positive Example

In the study of financial fraud discussed in this paper, TN denotes the number of samples in which non-financial fraud was correctly identified as non-financial fraud, FP denotes the number of samples in which non-financial fraud samples were identified as fraud, FN denotes the number of samples in which financial fraud samples were identified as non-fraud, and TP denotes the number of samples in which financial fraud was correctly identified as financial fraud.

In this work, SecureBoost, LNN and DPP-FedMeta are used to test the dataset respectively. These models were trained for 1000, 200 and 100 rounds with a batch size of 32. the user target privacy budget for DPP-FedMeta was set to 2. The model training process was stopped when the target privacy budget was reached. The cut value is set to 0.01 in the experiment and its used as the noise sensitivity. The amount of noise added to each round of training gradient by the noise scale. The larger the control noise scale is, then a larger amount of noise is added to the original gradient, which reduces the utility of the model, and the noise scale is set to 4 in the experiment. The comparison results of the three models are shown in Table 3.

The results show that the DPP-FedMeta model performs the best on the Precision metric at 93.99%, followed by the SecureBoost and LNN models at 92.31% and 90.51%, respectively. This indicates that the DPP-FedMeta model is more accurate in determining instances of financial fraud. According to the tabular data, the DPP-FedMeta model shows better performance under all the metrics and has better overall performance and performance in the financial fraud identification task compared to SecureBoost and LNN models.

Table 3. Comparative results of models/%

	SecureBoost	**LNN**	**DPP-FedMeta**
Precision	92.31	90.51	93.99
Recall	93.28	91.45	94.94
F1	92.27	91.32	94.46
Accuracy	92.56	91.25	94.48
AUC	98.07	93.71	98.58

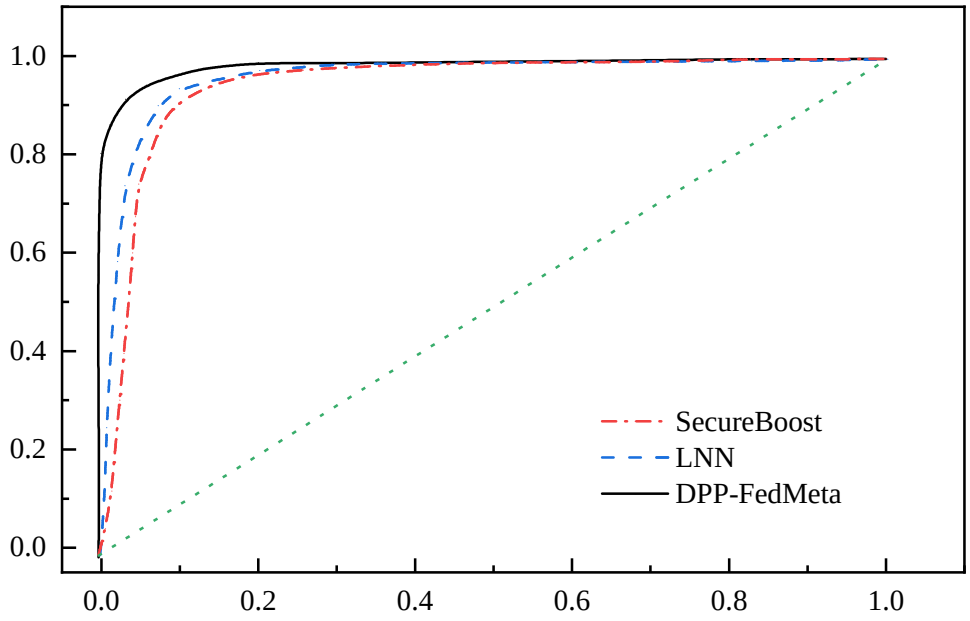

Figure 5. ROC curves

From the ROC curves as shown in 5, the DPP-FedMeta model has a steeper curve and covers a larger lower area than the other two models. This indicates that with the introduction of differential privacy protection, the FedMeta model is more decisive in its judgement and less likely to have fraud/non-fraud samples reasoning out probabilities in the same interval.

5. **Conclusion.** Based on FedMeta, a federated meta-learning framework, this paper proposes a differential privacy-preserving federated meta-learning method, DPP-FedMeta, which is utilised to construct a classification and identification model for financial fraud. The model objective is to train the metamodel collaboratively using data distributed among multiple clients. In addition, a federated meta-learning method based on differential privacy preservation is designed to protect the user's privacy security, which is more applicable to the financial fraud recognition model. Experimental results show that the DPP-FedMeta model exhibits better performance in Precision, Recall, F1, Accuracy and AUC metrics compared to SecureBoost and longitudinal neural network models.

Acknowledgment. This work is partially supported by the Key Project of the Guangxi Ethnic Regions Cultural Development and Social Governance Research Center (No. 2018YJJD0004).

REFERENCES

[1] J. M. Karpoff, "The future of financial fraud," *Journal of Corporate Finance*, vol. 66, 101694, 2021.

[2] J. West and M. Bhattacharya, "Intelligent financial fraud detection: a comprehensive review," *Computers & Security*, vol. 57, pp. 47-66, 2016.
[3] E. M. Fich and A. Shivdasani, "Financial fraud, director reputation, and shareholder wealth," *Journal of Financial Economics*, vol. 86, no. 2, pp. 306-336, 2007.
[4] H. A. Hashim, Z. Salleh, I. Shuhaimi, and N. A. N. Ismail, "The risk of financial fraud: a management perspective," *Journal of Financial Crime*, vol. 27, no. 4, pp. 1143-1159, 2020.
[5] E. W. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision Support Systems*, vol. 50, no. 3, pp. 559-569, 2011.
[6] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial fraud: a review of anomaly detection techniques and recent advances," *Expert Systems with Applications*, vol. 193, 116429, 2022.
[7] Z. Rezaee, "Causes, consequences, and deterrence of financial statement fraud," *Critical Perspectives on Accounting*, vol. 16, no. 3, pp. 277-298, 2005.
[8] M. Albashrawi, "Detecting financial fraud using data mining techniques: A decade review from 2004 to 2015," *Journal of Data Science*, vol. 14, no. 3, pp. 553-569, 2016.
[9] C. E. Hogan, Z. Rezaee, R. A. Riley Jr, and U. K. Velury, "Financial statement fraud: Insights from the academic literature," *Auditing: A Journal of Practice & Theory*, vol. 27, no. 2, pp. 231-252, 2008.
[10] K. G. Al-Hashedi and P. Magalingam, "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019," *Computer Science Review*, vol. 40, 100402, 2021.
[11] P. Richhariya and P. K. Singh, "A survey on financial fraud detection methodologies," *International Journal of Computer Applications*, vol. 45, no. 22, pp. 15-22, 2012.
[12] W. Zhou and G. Kapoor, "Detecting evolutionary financial statement fraud," *Decision Support Systems*, vol. 50, no. 3, pp. 570-575, 2011.
[13] D. V. Dooley, "Financial fraud: Accounting theory and practice," *Fordham Journal of Corporate & Financial Law*, vol. 8, S53, 2002.
[14] J. West and M. Bhattacharya, "Some experimental issues in financial fraud mining," *Procedia Computer Science*, vol. 80, pp. 1734-1744, 2016.
[15] M. DeLiema, M. Deevy, A. Lusardi, and O. S. Mitchell, "Financial fraud among older Americans: Evidence and implications," *The Journals of Gerontology: Series B*, vol. 75, no. 4, pp. 861-868, 2020.
[16] P.-F. Pai, M.-F. Hsu, and M.-C. Wang, "A support vector machine-based model for detecting top management fraud," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 314-321, 2011.
[17] K.-H. Shih, C.-C. Cheng, and Y.-H. Wang, "Financial information fraud risk warning for manufacturing industry-using logistic regression and neural network," *Romanian Journal of Economic Forecasting*, vol. 14, no. 1, pp. 54-71, 2011.
[18] S. Chen, "Detection of fraudulent financial statements using the hybrid data mining approach," *SpringerPlus*, vol. 5, no. 1, pp. 1-16, 2016.
[19] X.-G. Wu and D.-Y. Du, "An analysis on financial statement fraud detection for Chinese listed companies using deep learning," *IEEE Access*, vol. 10, pp. 22516-22532, 2022.
[20] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security & Privacy*, vol. 17, no. 2, pp. 49-58, 2019.
[21] S. Park, Y. Suh, and J. Lee, "FedPSO: Federated learning using particle swarm optimization to reduce communication costs," *Sensors*, vol. 21, no. 2, pp. 600, 2021.
[22] L. You, S. Liu, Y. Chang, and C. Yuen, "A triple-step asynchronous federated learning mechanism for client activation, interaction optimization, and aggregation enhancement," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24199-24211, 2022.
[23] T.-Y. Wu, J. Lin, Y. Zhang, and C.-H. Chen, "A Grid-Based Swarm Intelligence Algorithm for Privacy-Preserving Data Mining," *Applied Sciences*, vol. 9, no. 4, 774, 2019.
[24] Q. Mei, H. Xiong, Y.-C. Chen, and C.-M. Chen, "Blockchain-Enabled Privacy-Preserving Authentication Mechanism for Transportation CPS With Cloud-Edge Computing," *IEEE Transactions on Engineering Management*, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9757246
[25] L. Yang, J. Huang, W. Lin, and J. Cao, "Personalized federated learning on non-IID data via group-based meta-learning," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 4, pp. 1-20, 2023.
[26] F. Liu, M. Li, X. Liu, T. Xue, J. Ren, and C. Zhang, "A review of federated meta-learning and its application in cyberspace security," *Electronics*, vol. 12, no. 15, pp. 3295, 2023.
[27] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "Secureboost: A lossless federated learning framework," *IEEE Intelligent Systems*, vol. 36, no. 6, pp. 87-98, 2021.

[28] M. Ren, Z. Wang, and X. Yu, "Personalized federated learning: A Clustered Distributed Co-Meta-Learning approach," *Information Sciences*, vol. 647, 119499, 2023.

[29] F. Dong, X. Ge, Q. Li, J. Zhang, D. Shen, S. Liu, X. Liu, G. Li, F. Wu, and J. Luo, "PADP-FedMeta: A personalized and adaptive differentially private federated meta learning mechanism for AIoT," *Journal of Systems Architecture*, vol. 134, 102754, 2023.