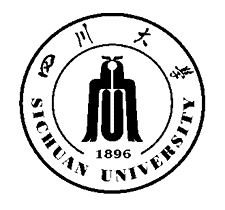


**本科生毕业论文（设计）**

****

**题 目 基于知识图谱的证券类金融问答系统设计**

**学 院 计算机学院**

**专 业 计算金融交叉试验班**

**学生姓名 杜林峰**

**学 号** 2019141460185 **年级** 2019

**指导教师 段磊**

**教务处制表**

**二〇二三年四月二十日**

基于联邦学习的信贷风控实证研究

专业：计算金融交叉试验班

学生：杜林峰 指导教师：段磊

摘 要 信用贷款风险控制简称信贷风控，在金融业中具有重要的地位。信贷风控一般指根据借款人的相关信息为其信用评级，并根据该评级的高低判断是否交易或放款。随着计算机技术的快速发展，结合计算机技术的金融科技已经用于信贷风控中的多个步骤，例如欺诈交易检测。但由于金融行业数据高度隐私、敏感带来的数据孤岛、数据隐私安全问题，结合传统机器学习方法的信贷风控无法获取多源数据用于风控建模导致风控模型效果欠佳。为了解决该问题，引入联邦学习进行风控建模。

联邦学习是一种起源于2016年的新型分布式机器学习架构，其强调“数据不动模型动，数据可用不可见”的特点能够较好地解决数据孤岛、数据隐私安全问题，使得全局风控模型能够获取多方的隐私金融数据，从而为机器学习模型提供更多的训练样本使其提升风控能力。

为了解决金融机构之间的数据孤岛问题，将联邦学习用于信贷风控中的欺诈交易检测步骤中，以实证联邦学习与信贷风控业务的可结合性。同时在实际金融场景中，风险交易（正样本）的数量要远小于正常交易（负样本）的数量，针对正负样本数量不平衡的问题，通过对比多种本地客户端的机器学习模型以及不同的过采样方法及比例，探究正样本过采样对全局模型带来的影响。在此基础上，提出了一种针对联邦学习中本地客户端过采样情况下的全局参数聚合方法FedAvgOS。该方法通过按Loss值的倒数降序排列并丢弃表现不佳的客户端模型参数，最后加权平均来进行全局模型参数聚合。经实验验证，FedAvgOS有效提高了联邦模型的收敛速度以及精度。

关键词 联邦学习；信贷风控；数据孤岛；机器学习

**A Study on Credit Loan Risk Control Based on Federated Learning**

Major: Computational Finance

Student: XIAO Yingjie Adviser: DUAN Lei

**Abstract** Credit loan risk control plays an important role in the financial industry. Credit loan risk control generally refers to the credit rating of the lenders based on the relevant information, and judging whether to trade or lend according to the level of rating. With the rapid development of computer technology, fintech combined with computer technologies has been used in many steps of credit loan risk control, such as credit fraud detection. However, due to the high privacy and sensitivity of financial data, data islands and data privacy security problems result in poor effect of risk control model with the traditional machine learning methods because of the lack of multi-source data. To solve this problem, federated learning is introduced into credit loan risk control modeling.

Federated learning is a new distribute machine learning architecture that originated in 2016. It emphasizes the characteristics of data privacy and security, which can better solve the data islands problems, enabling the global risk control model to obtain private financial data from multiple parties. Thus, more training samples can be provided for machine learning models to improve their performance.

In order to solve data islands and data privacy security problems, federated learning is used for detecting fraudulent transactions in credit loan risk control to demonstrate the possibility of combining federated learning and financial tasks. At the same time, the quantity of risk transactions(positive samples) is far less than the quantity of normal transactions(negative samples), aiming at the problem of unbalanced ratio of positive samples and negative samples, multiple machine learning modules of local clients and different oversampling methods with different ratios are compared to explore the influence of using oversampling methods bringing to the global model of federated learning. On this basis, a method of parameter fusion after oversampling in training steps for multi-client called FedAvgOS is proposed. In this method, the global model parameters are aggregated by arranging the parameters in ascending order of average loss’s reciprocal, then drops the parameters of the client models with poor performance, and finally gets the weighted average as the results of parameters for next round training. Experimental results show that FedAvgOS can effectively improve the convergence speed and accuracy of federated model.

**Key words** Federated Learning; Credit Loan Risk Control; Data Islands; Machine Learning

**目 录**

[第1章 绪论 1](#_Toc102997270)

[1.1 研究背景及意义 1](#_Toc102997271)

[1.2 主要研究工作 2](#_Toc102997272)

[1.3 本文组织结构 3](#_Toc102997273)

[第2章 相关工作 5](#_Toc102997274)

[2.1 基于传统统计学与机器学习方法的信贷风控建模技术研究 5](#_Toc102997275)

[2.1.1 基于监督学习的信贷风控建模 5](#_Toc102997276)

[2.1.2 基于无监督学习的信贷风控建模 6](#_Toc102997277)

[2.2 联邦学习相关技术研究 6](#_Toc102997278)

[2.2.1 联邦学习基本概念及分类 6](#_Toc102997279)

[2.2.2 联邦学习训练步骤 9](#_Toc102997280)

[2.2.3 联邦学习前沿技术研究进展 10](#_Toc102997281)

[2.2.4 联邦学习前沿应用场景 15](#_Toc102997282)

[2.3 本章小结 16](#_Toc102997283)

[第3章 信贷风控系统的分析与设计 17](#_Toc102997284)

[3.1 问题背景 17](#_Toc102997285)

[3.2 问题定义 18](#_Toc102997286)

[3.2 信贷风控系统重难点分析 18](#_Toc102997287)

[3.3 信贷风控系统各模块分析及总体设计 19](#_Toc102997288)

[3.3.1 数据预处理模块 20](#_Toc102997289)

[3.3.2 客户端模块 21](#_Toc102997290)

[3.3.3 中心服务器模块 22](#_Toc102997291)

[3.4 本章小结 22](#_Toc102997292)

[第4章 信贷风控系统的构建与实验 23](#_Toc102997293)

[4.1 系统开发环境 23](#_Toc102997294)

[4.1.1 开发工具 23](#_Toc102997295)

[4.1.2 开发语言及第三方框架/库/包 23](#_Toc102997296)

[4.2 信贷风控系统各重点模块的实现 25](#_Toc102997297)

[4.2.1 数据预处理模块 25](#_Toc102997298)

[4.2.2 客户端模块 27](#_Toc102997299)

[4.2.3 中心服务器模块 28](#_Toc102997300)

[4.2.4 本节常用符号汇总 32](#_Toc102997301)

[4.3 实验与结果分析 32](#_Toc102997302)

[4.3.1 实验环境 32](#_Toc102997303)

[4.3.2 实验数据分析 33](#_Toc102997304)

[4.3.3 实验评价指标 33](#_Toc102997305)

[4.3.4 实验结果分析 34](#_Toc102997306)

[4.4 本章小结 43](#_Toc102997307)

[第5章 总结与展望 44](#_Toc102997308)

[5.1 课题工作总结 44](#_Toc102997309)

[5.2 未来工作展望 44](#_Toc102997310)

[参考文献 46](#_Toc102997311)

[声明 50](#_Toc102997312)

[致谢 51](#_Toc102997313)

**图目录**

[图2-1 横向联邦学习 7](#_Toc101876884)

[图2-2 纵向联邦学习 8](#_Toc101876885)

[图2-3 联邦迁移学习 8](#_Toc101876886)

[图2-4 联邦学习典型训练步骤 9](#_Toc101876887)

[图3-1 基于联邦学习的信贷风控系统组织结构图 19](#_Toc101876888)

[图4-1 数据处理模块流程图 25](#_Toc101876889)

[图4-2 BorderlineSMOTE方法示意图 26](#_Toc101876890)

[图4-3 不同过采样比例下的联邦建模指标对比 38](#_Toc101876891)

[图4-4 不同客户端数量下的指标对比 40](#_Toc101876892)

[图4-5 FedAvgOS方法实验结果 42](#_Toc101876893)

**表目录**

[表3-1 本节符号汇总 18](#_Toc101876988)

[表4-1 系统开发环境 23](#_Toc101876989)

[表4-2 本节符号汇总 32](#_Toc101876990)

[表4-3 数据集信息表 33](#_Toc101876991)

[表4-4 混淆矩阵 33](#_Toc101876992)

[表4-5 本地客户端模型参数设置 35](#_Toc101876993)

[表4-6 本地客户端模型在不同过采样方法下的结果 36](#_Toc101876994)

[表4-7 本地客户端模型在不同过采样比例下的结果 37](#_Toc101876995)

[表4-8 不同参数聚合方法下的实验数据对比 41](#_Toc101876996)

**第1章 绪论**

1.1 研究背景及意义

随着计算机硬件算力的不断提高、软件算法的不断迭代更新，以及各行业相关数据的不断积累，人工智能（主要体现为机器学习）技术逐渐在各个行业中得到了较为广泛的应用。在金融领域，将传统的金融问题与新兴的深度学习、大数据、区块链等计算机技术结合的金融科技，已经成为了当下该领域最热门的话题之一。金融科技的蓬勃发展，既是对传统金融行业的一次巨大冲击，也是为其带来的一次重大机遇。如何将先进智能的计算机技术用于解决金融领域的传统问题，如股票走势预测、信贷风险控制等，从而提高金融业务的处理效率、金融客户的满意度等，已经成为了传统银行业、证券业、保险业、交易结算类等金融机构的首要目标。

信用贷款业务简称信贷业务，是银行业中最常见的放款形式，其地位在金融机构的业务之中尤为重要。信贷业务是指以偿还全部本金和利息为条件的价值转移方式，银行关注借款用户在规定期限内是否能够还本付息，因此需要对借款人的信用资质重重审核再进行放款。信贷风控本质上是通过对借款人的相关信息进行科学严谨地收集、统计、分析后，从而评估其在限期内还本付息的可能性。现在，机器学习方法（含深度学习方法）已经逐步取代了原来的基于简单统计方法的信用评分模型。机器学习方法通过输入借款人的各项相关信息，经过算法处理后，最后给出借款人的信用评分或是限期还款概率。然而，这些机器学习方法在直接应用于信贷风控建模时，存在一些问题或限制使其无法得到较为理想、准确的效果：

（1）数据孤岛问题。数据孤岛问题指一个行业内的各个企业，或是一个企业内的各个部门之间存在数据流通的壁垒，使得大环境内的有价值的数据因为无法交互共享而成为无价值、无意义的死数据。数据孤岛问题直接作用于基于机器学习的信贷风控建模过程上，反映为数据规模过小、有效数据不足或是数据特征不明显，从而导致模型效果欠佳。然而数据孤岛问题却无法直接解决，行业内部竞争、企业管理制度、数据隐私安全或是行政手续复杂等客观原因使得绝对意义上的数据整合往往是无法实现的，即便确实可行也会面临各方都无法承受的巨额成本。

（2）数据隐私安全问题。互联网在近二十年间飞速发展，数据大量积累，随之而来的网络隐私安全问题也越发突出，国内外都尤为重视。近年来，我国分别于2016年出台《中华人民共和国网络安全法》、2021年出台《中华人民共和国数据安全法》和《中华人民共和国个人信息保护法》，旨在以法促德，规范网络生态环境，切实保障数据隐私安全。同样在国外，欧盟于2018年出台《通用数据保护条例》（General Data Protection Regulation, GDPR）。在前不久的2022年1月，Google和Facebook就因未征得用户全部同意就强制跟踪用户浏览足迹从而违反《通用数据保护条例》被法国国家信息与自由委员会罚款2.1亿欧元[1]。由此可见，如何在数据的产生、使用过程中切实保护数据安全、个人隐私信息，已经成为了全球所共同面临的严峻挑战。

由于前述问题，如何在收集、统计以及利用有效的客户隐私数据以构建信贷风控模型的同时，有效降低数据孤岛问题带来的影响，同时保护借款人隐私信息，已经成为了当前风控领域的难点之一。针对数据建模的难点，Google在2016年首创联邦学习（Federated Learning, FL）技术，用于解决安卓终端的本地模型更新问题[2]。联邦学习是一种分布式机器学习框架，但又不同于传统的去中心化分布式结构，强调“数据不动模型动，数据可用不可见”[3]，对于数据孤岛问题、数据隐私安全问题有较好的解决效果。

在信贷风控领域，各家金融机构的私有数据高度敏感、高度隐私。如果各家机构只用自己的私有数据进行风控模型构建，会因为数据规模小、数据特征不明显等问题导致模型效果欠佳；如果各家打通数据壁垒，进行数据共享，虽然能构建出效果更好的风控模型，但几乎不能保证客户的隐私信息，以致丧失客户信任甚至引发大型金融刑事案件。联邦学习则为金融行业内的信贷风控模型构建提供了一种很好的思路。联邦学习使得各家金融机构的私有数据无需点对点交换或是统一上传到不可信的第三方数据整合机构，只需在本地训练后将本地模型的参数进行一定操作（如加密等）后上传至中心服务器，中心服务器通过相关算法融合后将新得到的模型参数下放至各客户端又开始新一轮的训练，直至模型效果达到某一阈值。这种训练模式一方面间接解决了数据孤岛问题带来的训练数据过少、数据融合成本过大的问题，另一方面也直接解决了金额数据高度隐私而无法共享的问题。综上所述，将联邦学习技术用于信贷风控模型建立具有良好的应用前景，因此实际证明其可结合性及有效性是一个非常具有实际意义的研究目标。

1.2 主要研究工作

本课题的研究目标是，将传统的信贷风控业务与联邦学习技术结合，在保证数据隐私安全的前提下，证明将联邦学习用于信贷风控模型构建是切实可行的、高效的。同时由于现实场景内发生金融欺诈交易、借款人无法在限期内还本付息发生的可能性较小，大部分的借款交易都是正常可信的，因此如何在正负样本数据比例悬殊的情况下达到较好的模型精度也是本课题的研究目标。具体地，主要研究工作如下：

（1）基于联邦学习分布式训练框架的实证研究

对现有的联邦学习技术进行归纳整理，同时对将联邦学习技术用于信贷风控模型构建中的重难点进行梳理归纳，以便更好地理解基于联邦学习的分布式训练架构构建流程、训练形式、底层原理，从而达到更好的模型精度、更高的运行效率。

（2）基于联邦学习的信贷风控模型设计与构建

本课题重点关注将联邦学习技术与传统信贷风控业务结合时所面临的重难点，如数据隐私安全、数据样本标签极度不平衡、单个借款人相关数据多而杂、数据特征不明显等问题，通过详细分析上述问题从而追求更佳的模型精度、更好的运行效率。通过特征工程等手段对原始数据进行处理后，将整个系统模块化、局部化，以软件工程“高内聚、低耦合”的思想来使模型各模块内部联系紧密、各模块之间分工明确，结构清晰，易于分析改进。

（3）基于联邦学习的信贷风控模型实现与验证

在对各模块逐一进行分析设计后，分别按照既定规划给出模块结构、算法框架，并进行编程实现。在各模块全部完成后，将所有模块进行组装从而形成完整的信贷风控模型，使用相关开源数据集对整个系统的可行性进行验证，并与传统的集中式本地机器学习模型进行对比，记录整理并对比相关指标以验证其性能。

（4）参数敏感性分析

针对联邦学习自身的技术特点，在验证联邦学习技术与信贷风控模型构建的可结合性后，对联邦学习训练过程中的相关参数进行敏感性分析，包括参与训练的本地客户端个数、过采样比例等。

（5）提高联邦学习中对少数类样本过采样后的全局模型收敛稳定性

由于金额风险交易数据正负样本比例悬殊，为提高系统对风险交易的识别能力，我们需要对少数类的风险交易样本进行过采样。但过采样方法几乎都是基于算法在一定范围内随机合成的，不同客户端过采样后的样本分布并不一致，在进行全局模型参数聚合时收敛较慢、模型精度不稳定。本文提出了一种针对联邦学习中本地客户端过采样下的全局模型的参数聚合方法FedAvgOS，使过采样后的全局参数聚合更加稳定、全局模型收敛更快。

1.3 本文组织结构

本文内容的组织结构如下：

第1章介绍了传统金融业务中信用贷款风险控制模型构建与机器学习方法结合所面临的难点，然后引入联邦学习技术以及明晰联邦学习技术用于信贷风控模型构建的优势与应用前景，最后整理出本文的实证研究目标以及行文组织结构。

第2章概述了传统机器学习方法用于信贷风控模型构建以及联邦学习技术相关的国内外研究工作，具体包括信贷风控领域内的传统机器学习方法以及所面临的难点、痛点；联邦学习的概述、分类以及本课题着重关注的金融数据训练下的联邦学习安全性以及通信效率。

第3章是对整个基于联邦学习的信贷风控模型的各个模块的详细分析与设计。从软件工程“高内聚，低耦合”思想出发，通过分析现有传统机器学习方法直接用于模型构建的难点，结合联邦学习技术，给出整个系统的结构图以及相关算法伪代码，用于指导具体的代码编程实现。

第4章具体描述整个信贷风控模型的实际构建过程以及对模型的测试结果。首先给出开发环境、开发工具以及相关框架、库、包，并对各模块的实现难点以及针对难点所提出的实现方案具体描述，着重体现面对前文主要研究工作中所描述的问题与挑战的解决手段。实验部分先给出相关数据集描述，并针对数据集特点分析特征工程要点；随后对模型进行详细的测试验证并分析其参数敏感性，并将本文提出的针对全局模型参数聚合的新方法与一般方法对比，证明其有效性。

第5章总结本文工作，对本次工作未涉及到的、有研究价值的点进行整理归纳，最后对今后的研究方向与内容进行展望。

**第2章 相关工作**

在联邦学习技术出现之前，银行业等金融机构的数据分析员、信贷业务员已经逐步完成了将信贷风控模型构建方法由传统统计学方法逐步转变为机器学习方法的过程。同时在2016年联邦学习被Google提出后，该训练模式引起了学者的广泛兴趣，有不少知名学者如杨强教授等对其进行了模型完善、改进。在此基础上，诞生了几个较为知名的联邦学习框架，它们有各自的优势与不足，其使用已经逐步由学术界转向工业界。本章将从传统的信贷风控建模构建出发，归纳总结信贷风控模型构建的发展与现状；同时对联邦学习详细介绍其工作原理、模型分类，以及当今所面临的挑战与问题。

2.1 基于传统统计学与机器学习方法的信贷风控建模技术研究

一般而言，信贷风控模型的构建主要是基于统计学习（Statistical Learning）方法。统计学习方法包括基于数理统计的传统统计学方法（如线性判别法等）以及现代机器学习方法。统计学习方法可以分为监督学习、半监督学习、无监督学习以及强化学习[4]。而一般的信贷风控模型主要是基于监督学习（包含深度学习方法）和无监督学习构建，以监督学习为主。

2.1.1 基于监督学习的信贷风控建模

监督学习（Supervised Learning）是指从有标签的数据中，学习输入到输出（标签）的映射规律，反应到信贷业务中就是对借款人的身份分类，即“好客户”和“坏客户”，或是“信用好”和“信用坏”。Durand[5]最早将线性判别分析法（Linear Discriminant Analysis, LDA）模型应用到信贷评估上，自此银行业的信贷业务分析由笼统的定性分析转为了具有强可解释性的定量分析上。但该方法在非高斯分布样本的数据集上效果并不好，同时它的分类依据是方差，在有异常点或噪声存在的情况下，分类效果并不好。Orgler[6]最早将线性回归（Linear Regression）方法用于信贷风险评估，通过对借款人的相关数据回归分析最终得出一个违约概率，并人为制定标准为概率按照大小归类。Orgler在他的实验中成功证明了消费者实际的行为特征比他们在借款时所填写的申请表信息更能反映他们的经济实力。但线性回归要求输入自变量为连续性数值变量，并且输出范围为()，为了解决这些问题，逻辑回归（Logistic Regression）被引入到信用评估模型中。逻辑回归允许输入变量为离散型类别变量，并且通过Sigmoid函数转化为二分类或者Softmax函数转化为多分类问题。Wiginton[7]等首先将逻辑回归应用于信贷风控上，此后Grablowsky[8]、Ziari[9]等人分别通过调整似然函数、组合线性回归等方式改进了逻辑回归模型的性能。Bian Yiyang[10]通过融合Bagging和Boosting技术提出了一种信用卡交易欺诈检测的模型。反欺诈是信贷风控体系的重要环节之一，旨在拒绝个人信用评估风险较大的借款人、贷款人所提出的交易请求。在该模型中，Bagging和Boosting技术分别被用于原始数据重采样和正负样本不均衡带来的分类方差。近年来，随着金融数据的大量积累，加之神经网络模型在近20年间于学术界与工业界得到了大量的应用，多种新型神经网络如卷积神经网络、长短期记忆神经网络等网络结构也被用于信贷风控模型的构建。Odom[11]等人首先在个人借款信用评估上引入感知机-人工神经网络模型，虽然这种模型对应为其中的激活函数对非线性数据评估效果好，但囿于对数据的巨量需求，所以直到2010年前后因为数据的大量积累和设备算力的提升才重新成为研究热点。Cuicui Luo[12]等发现深度信念网络（Deep Belief Network, DBN）相较于逻辑回归模型和支持向量机（Support Vector Machine, SVM）两种传统机器学习模型，在信贷风控上有更好的效果。此外还有不少神经网络模型都在信贷风控上取得了不错的效果。

2.1.2 基于无监督学习的信贷风控建模

无监督学习指用不带标签的数据进行训练，其通过分析各维度上数据的特征差异或关联来进行客户聚类，达到反欺诈等目的。邹雅莹[13]构建了一种基于k-means聚类算法的马田系统模型，以聚类的正确率信噪比作为特征并以此来评价筛选后的特征指标，取得了较好的效果。Chougule等人[14]将k-means聚类算法与遗传算法相结合，提升了交易欺诈检测的正确率。Tran[15]等人首先将数据进行主成分分析（Principal Component Analysis, PCA），转换为线性无关变量所构成的数据，随后在此基础上建立多分类回归模型。Zaslavsky[16]等人利用银行等金额机构的新数据以流水形式进入银行的特点，提出了一种名为SOM（Self Organizing Map）的无监督学习模型，该模型不需要先验标签信息，只需使用流水添加的新数据来更新模型。此外，无监督学习中的马尔科夫链、蒙特卡罗法等也曾被用于信贷风险评估。

2.2 联邦学习相关技术研究

2.2.1 联邦学习基本概念及分类

联邦学习是利用分散在各参与方的数据集，通过隐私保护技术融合多方数据信息，协同构建全局模型的一种分布式机器学习训练方式。设当前有*N*位数据拥有者参与联邦学习，第*k*位数据拥有者记为*Ck*，其数据集分为*Dk*。传统的集中式训练需要所有拥有者持有的数据*DN* = {*D*1, *D*2, ..., *D*N}上传至中心服务器，由中心服务器全权接管所有参与方的私有数据。在这种该情况下，数据互相可见，且无法防范中心服务器对全局或部分数据可能存在的隐私泄露或恶意使用。

下面给出联邦学习相对集中式训练的性能损失。设*VFED*和*VSUM*分别为联邦学习模型和集中式模型的性能度量（如准确率），有：

**定义2-1 联邦学习性能损失**[17]**：**

|  |  |  |
| --- | --- | --- |
|  |  | (2.1) |

则称联邦学习模型在允许的性能损失范围内达到了-精度损失。通常来说，是一个较小的浮点数（如= 0.01），在实际落地的应用中，会根据业务场景的不同调整值。当业务的隐私性要求较低，精度要求较高时，要求值较小，以达到最接近集中式训练的精度效果；当业务的隐私性要求较高，精度要求较低时，值会相应增大，以更大的损失精度换来更高的隐私安全性。

下面介绍联邦学习的分类。用户数据*Dk*通常以矩阵张量的形式存在，则每一个*Dk*可划分为样本ID空间、特征空间以及标签空间。根据各用户参与联邦学习数据的特征空间与样本ID空间的重叠关系不同，联邦学习可以分为横向联邦学习（Horizontal Federated Learning, HFL）、纵向联邦学习（Vertical Federated Learning, VFL）以及联邦迁移学习（Federated Transfer Learning, FTL）。

首先是横向联邦学习。横向联邦学习适用于特征空间重叠较多而样本ID空间重叠较少的情况。设有两个用户*A*、*B*各自有一个数据集，则有：

|  |  |  |
| --- | --- | --- |
|  |  | (2.2) |



**图2-1 横向联邦学习**

**Figure 2-1 Horizontal Federated Learning**

横向联邦学习如图2-1所示。例如两家分布于不同地区的银行都有几乎相同的信贷业务，它们所持有的借款人信息因为相似的业务模式而具有相似的特征空间。在这种情况下，不同机构的样本ID空间重叠较小，特征空间重叠较大，两家银行就可联合建立横向联邦学习模型进行信贷风控。

然后是纵向联邦学习。横向联邦学习适用于特征空间重叠较少而样本ID空间重叠较多的情况。设有两个用户*A*、*B*各自有一个数据集，则有：

|  |  |  |
| --- | --- | --- |
|  |  | (2.3) |



**图2-2 纵向联邦学习**

**Figure 2-2 Vertical Federated Learning**

纵向联邦学习如图2-2所示。例如两家分布于相同地区但提供不同服务的银行，它们的客户人群基本一致或相似，但所持有的客户数据特征存在较大差异。在这种情况下，不同银行本ID空间重叠较大，特征空间重叠较小，两家银行就可建立纵向联邦学习模型，以获得相似客户的更多数据特征信息，从而获得更好的风控能力。

最后一种是联邦迁移学习。联邦迁移学习则适用于用户数据的样本ID空间和特征空间都重叠较小的场景。设有两个用户*A*、*B*各自有一个数据集，则有：

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |



**图2-3 联邦迁移学习**

**Figure 2-3 Federated Transfer Learning**

联邦迁移学习如图2-3所示。联邦迁移学习一定程度上也适用于金融场景的风控建模。近年来由于新冠疫情等各种因素，小微企业借款难、生存难，国家相关监管机构因此出台近一步降低小微企业融资借贷成本的政策。但小微企业往往成立时间短，在信贷业务应用中存在数据稀缺、不全面、历史信息沉淀不足等问题。针对此类问题，可以利用联邦迁移学习将金融机构在中大型企业中应用的信贷模型迁移到小微企业中，帮助提供信贷风控效果。

2.2.2 联邦学习训练步骤

联邦学习有两种架构，分别是对等网络拓扑架构和用户-服务器架构（集中式拓扑）架构。在对等网络拓扑架构下，不存在中心服务器，各训练参与方的通信、训练地位相等。这种没有中心服务器的架构避免了“半诚实”或“恶意”中心服务器带来的潜在威胁，相比用户-服务器架构更加安全；但同时该架构的设计、实现难度较大，如何在实际各方通信、训练的过程中保证各方参与方地位相等并及时排除数据质量低、训练效率差的参与方，相关学者还正在研究。目前联邦学习采用的架构一般是用户-服务器架构，后文如果没有特殊说明，所指联邦学习都默认为用户-服务器架构。下面说明联邦学习的典型训练步骤。



**图2-4 联邦学习典型训练步骤**

**Figure 2-4 Typical Training Steps of Federated Learning**

如图2-4所示，联邦学习的典型训练过程分为四个步骤：

步骤①：各参与方从中心服务器下载最新的模型参数，并更新本地模型。但通常这一步并不是所有客户端都会参与，因为各客户端状态并不时刻一致，如最初用于安卓客户端更新本地模型的联邦学习中，一般是在安卓手机空闲且无较大网络流量占用时才会参与联邦学习训练。

步骤②：各参与方利用更新后的模型训练自己的私有数据集，训练完成后将参数、梯度更新等相关信息通过加密方式（如同态加密、差分隐私等）上传至中心服务器，中心服务器按照既定算法聚合各客户端上传的信息。

步骤③：中心服务器将聚合后的新模型参数下发给符合条件的客户端。

步骤④：各客户端更新本地模型后继续开始新一轮的训练。

上述步骤为典型的同步联邦学习步骤。在同步的联邦学习机制中，各客户端的更新机制是同步的。但在实际的应用场景中，各客户端的状态不一，设备的算力、网络的状况都不尽相同，很难做到同步更新、训练。Xie等人[21]提出了一种异步联邦学习方法，用于缓解联邦学习算法在同步问题上的额外开销。实验证明，对于强凸问题和非凸问题以及有限类非凸问题，这种异步的联邦学习算法都具有接近线性收敛的全局最优解。异步的联邦学习虽然在时空开销上有一定优势，但典型的同步式联邦学习将每一步骤模块化、统一化，使得在实际应用场景中更容易对联邦学习架构根据业务需要做出调整，如金融数据高度隐私、高度敏感下的金融信贷风控场景。我们在同步式联邦学习架构中，可以自行对其安全性做出调整，以适应业务场景的需要。但同时，这种典型的联邦学习训练步骤也有一些值得研究的点，例如如何根据客户端的质量（如数据质量、设备算力、是否有恶意等）来对参与联邦学习的客户端进行选择、如何将其与区块链及大数据等技术相结合以达到更好的训练效果、如何完善对客户端的激励机制，这些都具有现实的研究意义。

2.2.3 联邦学习前沿技术研究进展

联邦学习相对于传统的集中式机器学习的主要优势在于其对于数据隐私安全保护的优越性能。但联邦学习并不是完美的，自2016年联邦学习概念首提后，有部分联邦学习领域内急需解决的痛点与挑战逐渐被研究者发现，例如通信效率较低、网络负载大、隐私安全仍然存在缺陷、对Non-IID数据收敛效果差、模型异构性较大等，本节将对上述问题简述。

**（一）通信效率**

在联邦学习中，需要不断重复本地客户端与中心服务器之间的数据交换过程，以联邦学习最初用于安卓终端的本地系统更新为例，数以万计的客户端与服务器进行异步数据交换，存在着如网络不稳定、延时重连等网络问题，同时也会影响手机应用的正常使用。通常，设联邦学习所消耗的时间为*T*，则全局模型的训练时间可用下式表示：

|  |  |  |
| --- | --- | --- |
|  |  | (2.5) |

随着设备算力的不断提升，通信传输的时间逐渐成为了全局模型训练消耗时间的瓶颈。因此，提升通信效率具有非常高的研究价值。目前，主要有三种方式用于提高通信效率：（1）算法优化；（2）参数压缩；（3）分散训练。下面将简述这三种方法。

（1）算法优化。不同于提高网络带宽从而提升通信开销，算法优化主要通过通信方式的改进，使其在海量数据、多客户端、高频率、低容错的情况下更好地实现联邦学习的通信开销。Chen等人[22]最先在分布式系统上提出FedSGD算法，这也作为后来联邦学习通信的基线算法。McMahan等人[2]随后提出了经典的FedAvg算法，其原理是在本地客户端多次执行随机梯度下降（Stochastic Gradient Descent, SGD）算法后再与中心服务器进行参数交换。经过实验验证，在训练不同神经网络达到相同精度的通信轮数上最高减少了10%。由于FedAvg效果明显优于之前的基线算法，所以此后FedAvg开始成为联邦学习的通信基线算法。但FedAvg对于难以保证非凸问题及Non-IID数据集的收敛效果，因此在其基础上学者们开展了大量研究。Wang等人[24]提出FedMA算法，通过优化本地客户端模型训练效果从而减少额外开销，加速全局模型收敛速度。Reddi等人[25]将自适应算法融入到联邦学习通信中，通过构造自适应优化器提出了FedAdagrad、FedYogi和FedAdam三种通信算法，有效提高了联邦学习对于异构数据和非凸问题的收敛效果。此外还有一些工作是通过在本地客户端或中心服务器上增加筛选算法以降低通信数据的冗余性，使得对于网络的利用率更高，从而提高收敛速度。

（2）参数压缩。除了通信算法的优化外，还可以通过缩减通信数据规模的大小进而减少通信开销。目前较为流行的压缩方案有两种，分别是全局模型压缩和梯度信息压缩。相比之下，梯度信息压缩对通信效率作用的影响更明显，因为通常网络的上行速度是远小于下行速度的，梯度信息上传至中心服务器会消耗大量时间。Konečný等人[26]提出了一种针对本地模型的通过量化、子采样或随机旋转等操作进行更新的草图更新算法。上述方法属于有损压缩，即客户端与中心服务器之间的数据交换并非完整、正确的信息，其在随机梯度下降上的收敛速度会明显降低。Caldas等人[27]利用上述方法压缩全局模型的更新，并将神经网络中常用的Dropout思想引入联邦学习中，每次随机选择全局模型的子集并采用子采样等压缩操作，客户端将压缩模型解压后再进行训练，一定程度上减少了通信开销。但上述工作并未考虑数据异构的情况，有关适配真实联邦学习场景的压缩算法还需要进一步研究。

（3）分散训练。分散训练的结构类似一般分布式训练的结构，客户端通过近邻算法只与自己的邻居通信，更新模型信息。He等人[28]提出了一种联邦学习的分散式通信方式，在相关数据集上证明了分散式训练实现了通信效率、可伸缩性以及对数据变化的抵御能力，并且允许不可靠设备和异构数据的参与。然而真实训练场景中，分散式训练并不常见，如何保证每一个客户端公平地参与训练并且如何将模型更新信息进行全局广播仍然需要继续研究。

**（二）数据隐私安全**

数据隐私安全是联邦学习中另一个值得关注的方向。虽然联邦学习的机制有“数据可用不可见”的特点，但该特点仅在本地客户端成立。如果中心服务器存在“恶意”，那么本地客户端的私有数据可能被中心服务器上更新后的参数反推得到，即“推理攻击”、“投毒攻击”；如果任意本地客户端存在恶意，那么通过该客户端训练得到的模型参数就会与其他正常客户端训练得到的模型参数形成巨大差异，从而对全局模型的收敛造成巨大干扰。以上安全威胁对联邦学习的发展造成了巨大的阻碍，所以往往还需要更加保险、安全的方法为联邦学习提供额外保护。目前，同态加密、差分隐私、安全多方计算已经成为了保护数据隐私安全的主流技术，下面分别对其介绍。

（1）基于同态加密的安全机制

同态加密（Homomorphic Encryption, HE）最初由Rivest等人[29]于1978年提出。同态加密允许直接对明文加密后形成的密文直接输入机器学习模型并输出加密结果，无需对密文进行解密就可以直接处理密文。也就是说，客户端计算方法无需了解明文，同时又能对加密后的密文进行如加减乘除等运算操作，同时不会泄露任何原文内容。鉴于这些优点，同态加密目前已经成为联邦学习中最常用的数据隐私保护机制。设*H*是一种同态加密方法，下面是同态加密的定义：

**定义2-2 同态加密**

|  |  |  |
| --- | --- | --- |
|  | *H* *=* {*KeyGen, Enc, Dec, Eval*} | (2.6) |

其中，*KeyGen*是密钥生成函数（包含加密公钥、解密私钥），*Enc*是加密函数，*Dec*是解密函数，*Eval*是评估函数，用于验证加密算法的正确性。同态加密允许在密文上进行某些代数运算，以加法同态运算或乘法同态运算，设*u*和*v*是明文空间中的任意两个元素，则有：

|  |  |  |
| --- | --- | --- |
|  | *Encpk*(*u*) = *u*, *Encpk*(*v*) = *v*, *Encpk*(*uv*) = (*uv*) | (2.7) |
|  | *Decsk*(*uv*) = *Decsk*((*uv*)) = *uv* |

其中，*pk*和*sk*分别为加密公钥、解密私钥，为加法或乘法运算。如果只单独满足加法或乘法的一种运算，则被称为部分同态加密（Partially Homomorphic Encryption, PHE）；如果在满足部分同态加密的基础上，只能执行有限次数的操作运算，则被称为些许同态加密（Somewhat Homomorphic Encryption, SHE）；如果既满足加法也满足乘法运算且不限制运算次数，则被称为全同态加密（Fully Homomorphic Encryption, FHE）。

（2）基于差分隐私的安全机制

差分隐私由Dwork[30]于2006年首次提出。差分隐私将随机生成的噪声按照一定方式加入到数据中，使数据传输中的信息不完全正确，以防止潜在攻击对象进行推理攻击以获取个体敏感信息。差分隐私被广泛认为是当前基于微扰的隐私保护技术中安全性能最高的之一[31]。下面给出差分隐私的定义：

**定义2-3** **-差分隐私** 对于只存在一处记录不同的两个数据集和，以及一个随机算法，以及对于任意的输出*O*Range()，当且仅当：

|  |  |  |
| --- | --- | --- |
|  |  | (2.8) |

我们称随机算法提供**。**式中，Pr为算法的输出概率，表示隐私预算（privacy budget），其值的大小与隐私保护性能成反比关系，与计算开销成正比关系。表示失败概率，的大小与隐私保护性能成反比关系。差分隐私中目前广泛流行三种噪声机制，我们将在后文中介绍。

（3）基于安全多方计算的安全机制

1982年姚期智[32]针对“百万富翁问题”提出安全双方计算，并由Goldreich等人[34]于1987年推广至安全多方计算。设共有*N*个参与者，表示每个参与者参与协同计算中所携带的私有内容，表示经过安全多方算法*f*后所得到的对应输出信息，则安全多方计算可用下式表示：

|  |  |  |
| --- | --- | --- |
|  | *y1*, *y2*,..., *yN* = *f*(*x1*, *x2*,..., *xN*) | (2.9) |

当前主要有三种计算框架[35]实现基于安全多方计算的安全机制，分别为秘密共享（Secret Sharing, SS），不经意传输（Oblivious Transfer, OT）以及混淆电路（Garbled Circuit, CC）。

（4）三种安全机制的性能对比

上文介绍了联邦学习目前主流的三种数据隐私安全保护机制，它们起源于密码学，在与联邦学习的结合中各有优点和不足，我们需要根据实际情况来确定采用哪种安全机制。下面就计算性能、通行效率以及本地数据安全性三个方面来讨论三种安全机制的优点与不足。因为安全多方计算有很多种形式，这里以秘密共享为例。

* 计算性能：计算的耗时主要体现在模型更新时，采用随机梯度下降或自适应时刻估计方法（Adaptive Moment Estimation, Adam）等参数更新方法中的梯度求取上。同态加密方法的计算是在秘钥加密后的密文上进行，因此耗时会较差分隐私和安全多方计算更多，而后两者之间因为是在明文上进行计算，所以二者的耗时并无较大差异。
* 通信效率：通信效率主要由数据传输的次数和单次传输数据的大小决定。从数据传输的次数来看，同态加密和差分隐私基本一致，无需分步传输；而秘密共享通常会将单批数据拆分并多方、多向传输，因此其传输次数会多于另外两种方法。从单次传输的数据大小来看，差分隐私与秘钥共享传输的都是明文数据，所占比特数较小；而同态加密传输的密文数据所占比特数较大。综上所述，同态加密的通信效率最低，其次是秘密共享，差分隐私的通信效率最高。
* 综合安全性：在联邦学习中，全局模型的训练体现为本地模型与中心服务器关于模型的参数交换，而非原始数据。但目前有研究表明，“半诚实”的中心服务器或恶意第三方甚至能够通过交换的参数或梯度信息来反推原始数据[36]。在三种安全机制中，同态加密由于传输的是密文数据，如果接收方没有解密私钥的话几乎不能破解，所以其安全性最高；秘密共享因为在单批数据拆分、多方发送的过程中，存在多个恶意客户端“串通合谋”从而进行数据交换的可能性，因此安全性次于同态加密；差分隐私的安全性由添加在其中的噪声决定，噪声大小与安全性成正比，与模型性能成反比。一般而言，噪声不宜添加过大，以防止模型性能达不到既定要求。

综上所述，三种安全机制各有优点与不足。差分隐私虽然在安全性上有一定损失，但考虑到本课题所选用的都是自建虚拟客户端，不存在恶意第三方或“半诚实”中心服务器；同时差分隐私计算性能高、通信效率高，因此本课题采用差分隐私作为数据隐私安全机制，具体的步骤及算法将在后文说明。

**（三）异构性**

在联邦学习中，异构性指一个系统内的组成设备、参与方式不尽相同。例如最先用于安卓终端系统更新上的联邦学习，有的设备是在3G网络下参与联邦学习，有的设备是在Wifi网络下参与联邦学习；有的设备性能先进、算力较高，有的设备性能较差、算力较低。具体而言，联邦学习中的异构性可分为数据异构性以及系统异构性。

（1）数据异构性

在前文已经提及非独立同分布数据（None-Independently Identical Distribution, Non-IID），即数据不都服从一个隐含未知的统计分布，并且数据不是完全独立地从该分布上采样得到的。在联邦学习中，各客户端参与联邦学习的私有数据的数量和形式都不完全相同，而传统机器学习一般要求数据独立且同分布，以使得经过训练集训练后的模型能最大程度上地适配测试集。当然机器学习并不总是要求数据符合独立同分布，已有不少研究开始着手于Non-IID数据，如联邦学习领域。Wang等人[39]提出了Favor框架和基于deep Q-learning的学习机制，有效平衡了Non-IID数据带来的偏差并加速收敛；与FedAvg算法相比，通信轮数分别在MNIST、FashionMNIST、CIFAR-10共3个数据集上减少49%、23%、42%。Mills等人[40]提出了基于分布式Adam优化和压缩技术的CE-FedAvg算法，经过实验验证，在Non-IID数据上表现良好，达到指定准确率的时间仅为FedAvg的0.59倍。此外还有学者提出的一些通信算法[41]能有效减少Non-IID训练数据引入的偏差，在收敛速度、非凸问题解决上优于FedSGD等基线算法；但与此同时，如何在解决数据异构性的同时尽量降低模型复杂度以及通信、存储的开销，仍然需要继续研究。

（2）系统异构性

系统异构性指参与联邦学习的客户端之间由于硬件配置（包括设备算力、网络带宽、处理器架构、存储能力等）差异所带来的异构性。例如，在一批参与联邦学习的客户端中，有的客户端算力明显高于其它设备，按照现有联邦学习的激励与信任机制，最后的模型效果可能会偏向该类客户端所训练的本地模型效果。除了公平性问题外，如果某客户端存在恶意或不完全可信，那么它可能倾向提高算力从而使全局模型偏向自己预设的效果，从而引发联邦学习安全性问题。Hanzely等人[42]提出了L2GD和L2SGD+ 共2种联邦学习模型，强调部分设备的参与，将全局模型的随机梯度下降与本地模型的随机梯度下降混合，对异构数据有较好的处理效果。此外还有一些方法针对系统异构性[43]，但如何在通信成本、隐私安全成本与模型效果之间取得平衡还有待继续研究。

2.2.4 联邦学习前沿应用场景

不止金融领域，事实上在许多行业、许多领域内都存在严重的数据孤岛问题与数据隐私安全问题。如何在保障数据隐私安全的情况下，最大程度地进行数据共享，打破行业壁垒、最大化数据价值，联邦学习为此提供了一个可行的方案。本节将重点对未来联邦学习前沿应用可行性较高的3个场景进行分析介绍。

（1）金融信贷风控与保险定价

联邦学习在金融信贷风控、保险定价等金融相关领域内将会有较大的应用潜力。金融行业相较于其他行业往往会收到更严格、规模更大、覆盖面更全的法律管控，跨部门、跨企业、跨行业的数据一般是无法共享的。同时如前文所述，跨企业、跨行业的数据由于在数据库中没有统一的存储格式，例如在传统银行内部，客户经理更喜欢用Excel来整理数据，而移动运营商则一般通过JSON格式与第三方进行数据交互[44]。不同存储方式导致的异构性往往会导致全局数据特征缺失，使得本地单点建模的性能大幅下降。除了基于联邦学习的信贷风控建模外，金融领域还可以加大对联邦学习的运用，例如可以使用纵向联邦学习对保险产品进行个性化定价、使用横向联邦学习进行银行反洗钱建模[45]。

（2）个性化推荐

传统的推荐算法如协同过滤等已经较为成熟，其又可以通过关联算法、矩阵分解、神经网络、图模型等方式实现。个性化推荐包括视频推荐、商品推荐等，随着近几年短视频平台在国内外热度逐渐走高，将视频推荐与商品推荐结合，在较短时间通过视频等方式为用户推荐符合其需求的商品已经成为了各大互联网公司、电商平台的研究热点。一般而言，收集用户的数据越多，其对于视频、商品的推荐也就越准确。单一平台的用户数据规模较小，特征不清晰，此时联邦学习就可以在割裂的数据中进行全局建模，提升个性化推荐效果。

（3）智慧医疗

智慧医疗是当前计算机科学领域内一个热门的结合方向，在医疗领域内构建知识图谱、利用计算机视觉技术识别病患病征等结合方式已经在世界范围内得到大力推广。机器学习技术的出现极大程度地提升了医疗效率，降低了医疗人工成本。但患者相关数据高度隐私、特征维度很高，从单一的病征照片、血尿常规、基因序列等很难综合得出有效的治疗方案，且不同源的患者数据异构性较强、融合成本高，很难直接将不同方式描述的患者特征直接融合进行机器学习建模。联邦学习中的联邦迁移学习则为智慧医疗提供了一个很好的解决方案，联邦迁移学习无需直接融合各医疗源的私有数据，模型各自训练、全局共享，有效降低了各医疗源的私有数据宏观融合带来的数据异构性。

2.3 本章小结

本章首先介绍了基于传统机器学习方法的信贷风控建模方法，接着详细介绍了联邦学习的概念、分类、典型的训练步骤，随后介绍了当前联邦学习所面临的挑战与问题并给出与之对应的相关研究工作，最后介绍了3个未来联邦学习的代表性应用场景并给出解释。

**第3章 信贷风控系统的分析与设计**

本章将介绍传统的信贷风控建模流程，以及模块化的基于联邦学习的信贷风控系统构建。在本章的前两节中将会介绍在信贷系统构建过程中的详细步骤以及重难点分析，第3节介绍系统的各模块的功能与作用。

3.1 问题背景

本课题的主要目标是将传统的信贷风控业务与联邦学习技术结合，在保证数据隐私安全的前提下，证明将联邦学习用于信贷风控模型构建切实可行、高效安全。传统银行业的信贷业务体系往往由下述几个步骤或方向构成：

（1）数据收集

在客户申请信用贷款时，金融机构需要全面、科学地收集借款人和借贷产品的相关信息。其中借款人的数据可以分为外部数据和内部数据：其中，外部数据指借款人申请信贷产品时按照借贷机构要求所填写的信息，包括信用卡信息、存款信息等；内部数据指通过国家和行业相关机构所提供的信息，例如个人征信、犯罪记录等。借贷产品相关信息包括产品利率、还款期限、产品类型等。

（2）反欺诈交易

欺诈交易是在贷前对客户进行意图审查的步骤。1988年底，巴塞尔新资本协议[[1]](#footnote-0)将金融系统操作风险按照损失类型分为7类，其中就包含2项欺诈类型风险：内部欺诈，指由于金融机构内部人员利用程序漏洞或违反相应规章制度故意导致的损失；外部欺诈指第三方违法诈骗、盗用资产造成的损失。

一般公开数据集反映的都是外部欺诈，主要为信用卡欺诈，其中最常见的是不法分子窃取信用卡拥有者的信用卡信息并以此进行多方借贷，将借款全部转入自己账户内，令信用卡所有者在不知情的情况下被办理多种借贷业务，无力偿还导致征信受损。反欺诈交易主要是根据借款人的信息分析其本次交易是否带有欺诈意图，如果是，则拒绝掉并通知有关部门进行进一步审查。

（3）风控建模

信贷风控建模可以针对信贷业务流程中的某一个步骤进行建模，也可以针对全局综合建模。信用评分卡模型是风控领域内最常见的模型，它根据借款人的各项信息，借助一定算法最后给出借款人信用评分。一般而言，信用评分卡模型根据采用数据的时间点划分为贷前的申请评分卡（Application Score Card, ASC）、贷中的行为评分卡（Behavior Score Card, BSC）和贷后的催收评分卡（Collection Score Card, CSC）。本课题主要关注的是贷前的申请评分卡模型，重点关注信贷审批前的欺诈交易检测。

（4）还款催收

催收指借款人在邻近还款期限时还没有还款记录，金融机构将会通过人工催收、机器催收等方法催促借款人到期还款。如果借款人在还款期限到期后还未还清，则会执行对借款人的本次交易作坏账处理、将借款人加入信用黑名单、为借款人及本次信贷交易贴欺诈标签等操作。

以上4点为信贷风控流程中的关键步骤。在上述4个步骤的背景下，本课题选取其中的反欺诈步骤作为研究目标，实证联邦学习与信贷风控过程中反欺诈步骤的可结合性。

3.2 问题定义

本课题所研究的信贷风控主要关注其中的反欺诈交易部分。反欺诈交易是信贷风控体系内重要的组成部分，通过反欺诈交易从而拒绝潜在信贷风险是信贷风控体系的重要组成部分。从反欺诈交易的结果来看，其问题可以描述为一个二分类问题。通过分析借款人的各项数据，经过全局的联邦学习模型以及本地客户端的机器学习模型训练后，判断欺诈交易的类别。设第*k*个客户端的私有数据为，其特征空间为，标签空间为。对于第*i*条数据，= 1表示正样本，属于欺诈交易；= 0表示负样本，属于正常交易。表3-1列出了本节常用符号。

**表3-1 本节符号汇总**

**Table 3-1 Symbol Summary of This Section**

|  |  |
| --- | --- |
| **符号** | **含义** |
|  | *m*行*n*列的实数域 |
|  | 特征空间 |
|  | 样本标签空间 |
| *Dk* | 第*k*个客户端的私有数据集 |
|  | 第*k*个客户端的特征空间中的第*i*条数据所有维度数值构成的向量 |
|  | 第*k*个客户端的第*i*条数据对应的标签值 |

3.2 信贷风控系统重难点分析

在基于联邦学习的信贷风控系统中，针对现实场景建模而言，主要有以下重点、难点：

（1）正负样本数量倾斜

在真实信贷场景下，由于国家政策和相关法律法规的监管约束，真实存在的诈骗类风险交易实际是非常少的。信贷风控系统的目标是正确分辨风险交易与非风险交易，将风险交易作为正样本，非风险交易作为负样本。一般情况下，正负样本的比例十分不平衡，在某些取自现实银行或其他金融机构的真实数据集的比例甚至可达1 : 500。而机器学习模型，尤其是神经网络模型严重依赖样本规模，缺乏正样本的训练集使得模型无法从中较好地学习到正样本的特征信息，从而很难在真实信贷风控场景下进行对风险交易的甄别。

（2）选取正确的评价指标

由于正负样本的极度不平衡，在测试模型性能时需要格外注意测试的指标。传统的机器学习二分类测试指标有Accuracy、Recall、Precision等指标，但需要注意的是，在正负样本极度不平衡的数据集中，即使不通过机器学习模型，将所有测试集样本的预测结果全部记为*yi* = 0，即将测试集所有样本全部认为是负样本，整个信贷系统的准确率指标也会非常高，因为大量的负样本已经掩盖了正样本带来的损失。

（3）高并发、高性能

在实际的金融交易场景中，每秒的模型判断请求可能高达上百万次。而联邦学习又需要高性能、高带宽的网络连接作为基础，所以在满足模型精度的前提下，如何提高模型的稳定性、提高模型收敛速度就成为了基于联邦学习的信贷风控系统构建的重点之一。

3.3 信贷风控系统各模块分析及总体设计

本课题实现的基于联邦学习的信贷风控系统主要由三个模块构成，分别为数据预处理模块、客户端模块、中心服务器模块，分别对应整个系统中处理欺诈交易样本数据的功能、联邦学习各本地客户端操作（包含本地机器学习模型）、联邦学习中心服务器处理功能。系统组织结构如下图：



**图3-1 5基于联邦学习的信贷风控系统组织结构图**

**Figure 3-1 The Structure of the Risk Control System Based on Federated Learning**

3.3.1 数据预处理模块

数据预处理模块可分为两个小模块，分别为特征工程模块以及数据平衡模块，下面将逐一说明其作用。

（1）特征工程模块

特征工程是机器学习中至关重要的环节之一，对于机器学习模型的最终效果有决定性的影响。在本课题中，特征工程模块包含样本基本分析、清洗异常样本、处理缺失值、数据降维、数据标准化、数据格式转化等操作。

* 样本基本分析：分析样本大小、正负样本比例、数据类型等，为数据平衡模块提供理由支撑；
* 异常样本清洗：排除明显异常的离群样本点，避免对模型效果造成干扰；
* 处理缺失值：鉴于原始数据集数据规模较大，因此可以直接丢弃掉有特征缺失的样本；
* 数据降维：在信贷业务场景中，借款人的相关数值信息非常多，且并非完全独立，如果以原始特征空间作为数据输入直接进行训练，最后会导致模型收敛速度慢、收敛效果差等问题，因此需要进行特征降维。
* 数据标准化：由于样本中各特征的量纲不完全相同，所以需要对数据进行标准化然后进行分析，避免数据单位带来的影响。
* 数据格式转化：在不同机器学习框架下，需要对数据的格式进行转化。一般数据集中的数据为字符串格式，需要根据所选机器学习模型的不同将其进行例如字符串到浮点数、浮点数组到张量（tensor）等转化操作。

（2）数据平衡模块

前文已经提到，基于联邦学习的信贷风控系统中，各客户端的底层机器学习模型需要大量的正负样本同时训练，才能提高对其正确分类的能力。但一般而言，金融信贷风险交易发生的可能性较小，所以在网络上公开的数据集中，正负样本大多都非常不均衡。针对数据不平衡问题，一般有两种方法，分别是从数据角度出发的重采样（Resampling）方法以及从算法角度出发的分类代价差异性优化算法。前者可分为过采样（Over-sampling）、欠采样（Under-sampling）以及综合采样（即同时对原始数据进行过采样和欠采样）；后者主要是基于代价敏感学习算法（Cost-Sensitive Learning），代表算法为AdaCost算法[47]。目前多数方法主要是基于重采样方法，因为从数据角度入手更易实施，并且带来的时空开销要比算法优化低。因此数据平衡模块主要基于重采样方法实现。下面将对三种类型的重采样逐一说明：

* 过采样：按照一定算法规则或随机对原始数据样本中的少数类样本进行复制或合成，直至达到既定正负样本比例。其中随机过采样所选的多数类样本没有限制，尤其是随机复制少数类样本，极易造成模型过拟合。所以我们选择按照一定算法规则进行过采样，目前已有的工作如SMOTE[48]（Synthetic Minority Over-sampling Technique）、Borderline-SMOTE[49]、ADASYN[50]（Adaptive Synthetic Sampling）等过采样方法。
* 欠采样方法：按照一定算法规则或随机对原始数据样本中的多数类样本进行剔除，直至达到既定正负样本比例。但通常情况下，使用欠采样方法的机器学习模型往往分类效果不如未经过数据平衡处理的模型，因为欠采样方法会丢失很大一部分原始数据的信息，但模型实际预测的是与原始数据分布一致的测试集。所以本课题中并未考虑欠采样方法。
* 综合采样方法：对原始数据中的多数类样本进行欠采样，对少数类样本进行过采样的综合方法。代表性算法有SMOTEENN[51]和SMOTETomek[52]。

鉴于真实信贷业务场景下，训练数据特征维度高、正负样本极度不平衡的情况，我们采用过采样方法进行数据平衡。这种方式既避免了原有训练数据的信息丢失，也能增加模型对正样本的辨识能力，提高分类器精度。

3.3.2 客户端模块

客户端模块主要作用是在本地训练机器学习模型，再通过中心服务器进行全局模型聚合，得到更新后的模型参数后又下发到各客户端开始新一轮的训练。客户端模块主要由以下3个部分构成：

（1）训练模块

训练模块最主要的作用是使用本地该客户端的私有数据进行本地机器学习训练。其底层为我们实现设定好的机器学习模型。为了验证联邦学习用于信贷风控领域的可行性，我们需要将传统本地集中式模型训练的效果指标与联邦学习场景下的分布式训练模型的效果指标对比。这一部分我们需要控制的变量为本地机器学习模型种类、模型超参数等。

（2）测试模块

测试模块需要用测试集对本地训练结束后的模型进行测试评价，记录并保存结果，得到模型在真实场景下的泛化误差。

（3）评价模块

在评价模块中，我们要为本地客户端训练好后的模型与中心服务器聚合后的模型都设置一个可解释、合理有效的评价指标。前文已经提到，传统的单一指标如Accuracy已经无法公正合理地评价一个正负样本比例及其失衡的数据集，因为即使全部预测为负样本，它的准确率也会很高。因此在评价模块需要设置一个较为合理、客户的评价指标用于评价机器学习模型效果。

3.3.3 中心服务器模块

中心服务器模块的主要作用是作为全局网络通信的枢纽，负责和各客户端之间进行网络通信、参数发送与接收，通过一定的算法融合全局模型并下放至各客户端开始新一轮的训练。中心服务器模块主要由以下2个部分构成：

（1）网络通信参数聚合模块

由于联邦学习参与设备众多，考虑到真实的业务场景，需要选择开销小、延时低、对异构数据有良好处理能力的通信算法来接收并通过对过采样后的样本效果好、收敛快的参数聚合算法来聚合各本地客户端的模型参数。

（2）数据安全模块

联邦学习中实现安全机制大致有3种办法：同态加密、差分隐私和安全多方计算。差分隐私计算性能高、通信效率高，虽然由于添加噪声会使得模型的分类性能略微下降，但考虑到其优点，我们将使用差分隐私作为信贷风控系统的安全方式，具体表现为在客户端每轮的梯度下降结束后，对参数进行裁剪；全局模型参数聚合时，添加噪声。具体实现方式将在第4章说明。

3.4 本章小结

本章对整个基于联邦学习的信贷风控系统进行了模块化的解构与分析。首先明晰课题的问题定义，在将问题拆分重组聚合的同时分析出建模重点、难点，并针对这些问题有条理地、模块化地，按照软件工程“高内聚，低耦合”的设计原则设计出了信贷风控系统最上层的3个大模块，每个大模块下面又分2 ~ 3个小模块。这样的设计使得整个风控系统条理清晰、逻辑清楚，便于后期做增删修改。系统各模块的具体实现详见第4章。

**第4章 信贷风控系统的构建与实验**

在第3章中，我们对基于联邦学习的信贷风控系统从整体到部分进行了详细的分析与设计，本章将在第3章的基础上，完成系统在实际开发环境中的搭建，并将系统与传统的本地集中式训练对比，检验联邦学习的有效性。

4.1 系统开发环境

系统开发环境的配置表如下：

**表4-1 2系统开发环境**

**Table 4-1 The Developing Environment Configuration of the Risk Control System**

|  |  |
| --- | --- |
| **项目** | **配置** |
| 操作系统 | Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-99-generic x86\_64) |
| 处理器 | AMD Ryzen Threadripper 3990X |
| 显卡 | Nvidia RTX 2080Ti |
| 开发工具 | Visual Studio Code |
| 开发语言 | Python 3.7.12 |
| 框架/库/包 | flwr 0.18.0, scikit-learn 1.0.2, torch 1.10.1等 |

下面对整个开发环境的软件配置作简要说明：

4.1.1 开发工具

Visual Studio Code是Microsoft公司于2015年推出的一款免费的轻量级跨平台源代码编辑器，有着强大的扩展功能和生态环境，可以通过安装插件的形式大大拓展其功能如远程SSH连接、网页实时渲染、支持近二十种开发语言编辑。选用Visual Studio Code使得系统能非常便捷地连接远程服务器，直接在本机上编辑服务器上的代码并实时运行，速度快、效率高。

4.1.2 开发语言及第三方框架/库/包

（1）开发语言

开发语言选用Python 3.7.12。Python是目前机器学习最常用的语言之一，它基于C语言实现。相较于C++ 等高级语言，虽然Python损失了部分性能，但它语法简单直观，容易被初学者所接纳。并且Python生态丰富，许多第三方机器学习库/框架都有Python的接口接入，例如TensorFlow、PyTorch、keras等。因为联邦学习中需要使用第三方框架来进行客户端与中心服务器的数据交互，所以选用Python作为开发语言。

（2）第三方框架/库/包

* flwr 0.18.0：flwr的全称是Flower[53]，它是由Daniel等人于2020年开发的一个联邦学习框架。它有以下几个特点：a)可扩展性。Flower对参与联邦学习的客户端数量没有限制，只取决于中心服务器的计算性能和网络带宽。b)兼容几乎所有机器学习框架，包括keras、TensorFlow、PyTorch等。c)兼容几乎所有类型设备，包括单机设备、移动设备、云设备等。d)具有工业部署能力。Flower具有强大的扩展能力，能从单机项目部署扩展到工业级项目部署。e)对异构设备支持良好。Flower与不同的操作系统和硬件平台可互相操作，在异构边缘设备环境中运行良好。f)简单易上手。Flower遵循“高内聚、低耦合”设计原则，封装程度高，简便易上手。据其官网显示，Flower已成为剑桥大学、牛津大学、清华大学、诺基亚贝尔实验室等知名高校与企业的联邦学习指定合作伙伴。Flower对异构设备支持强大，且支持多种联邦学习通信算法，允许在单机环境下模拟客户端与中心服务器，无需高质量网络带宽支持，因此选用Flower作为信贷风控系统的联邦学习框架。
* scikit-learn 1.0.2：Scikit-learn[54]起源于Cournapeau等人在2007年开发的Google Summer of Code项目，后经Pedregosa等人于2011年在机器学习顶级刊物JMLR上发表。Scikit-learn是全世界范围内最受欢迎的机器学习库之一，它以Python编写，广泛支持包含分类、回归、聚类等基本或高级机器学习算法，使用清晰直观，无需开发者手动编写大量机器学习底层代码。因此选用Scikit-learn作为联邦学习中客户端的机器学习算法框架。
* torch 1.10.1：torch的全称是PyTorch[55]，由Facebook人工智能研究院（FAIR）的Paszke等人开发完成，支持GPU加速下的张量计算以及便捷的神经网络开发。在联邦学习的本地客户端，我们会选用部分神经网络模型作为本地机器学习算法，因此PyTorch也作为第三方库支撑信贷风控系统中神经网络部分的开发。

4.2 信贷风控系统各重点模块的实现

4.2.1 数据预处理模块



**图4-1 6数据处理模块流程图**

**Figure 4-1 The Flow Chart of the Data Processing Module**

（1）特征工程模块

特征工程模块内我们重点关注数据标准化和数据降维：

* 标准化：由于样本中各特征的量纲不完全相同，所以需要对数据进行标准化然后进行分析，避免数据单位带来的影响。我们主要采用Z-score标准化方法：

|  |  |  |
| --- | --- | --- |
|  |  | (4.1) |

其中，*x*为某条样本中在某一维度上的值，为其均值，为其标准差。通过标准化后，我们就能避免量纲不一致对模型带来的影响。

* 数据降维：数据降维部分采用主成分分析方法，通过线性变换将原始数据转换为各维度线性无关的正交表示，不仅能提取数据的主要特征分量，还可以将高维数据转化为低微数据进行可视化。

（2）数据平衡模块

在第3章已经说明，基于联邦学习的信贷风控系统中，我们主要采用过采样方法平衡数据集正负样本分布。然而需要说明的是，样本的过采样并不是必须的，因为在真实的金融场景中，金融分析师普遍认为，取样自真实交易的不平衡数据集本身就蕴含着正负样本的分布信息，如果采用重采样方法强行平衡数据，可能会丢失数据集中蕴含的真实金融信息。所以是否采用过采样方法由具体的机器学习方法决定，以达到更佳的分类效果。

在数据平衡模块中，我们主要关注过采样方法以及采样率。采样率指的是使用过采样方法预期达到的正负样本比率，默认为1 : 1。在正样本的真实数量不大的情况下，我们更要谨慎选择采样率，防止对正样本的过量采样导致在训练集上对正样本过拟合导致测试结果不佳。

下面介绍两种较为常见的过采样方法：

* SMOTE：SMOTE是Chawla等人于2002年提出的过采样方法，算法的思想是合成少数类样本，它按照一种距离（如曼哈顿距离、欧氏距离等）对每个少数类样本*xi*从*K* (*K* = 1,2,...) 个近邻中随机选择*N*个样本*xzi*，然后依次在*xi*和*xzi*间随机合成新样本，公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (4.2) |

其中，为样本差异系数，随机使用保证对于每一个少数类样本及其近邻间合成的点在两点之间，并且距离随机。

* BorderlineSMOTE：BorderlineSMOTE是Hui Han等人于2008年提出的一种基于SMOTE的改进改进过采样方法。如图4-2所示：



**图4-2 7BorderlineSMOTE方法示意图**

**Figure 4-2 The Schematic Diagram of BorderlineSMOTE**

BorderlineSMOTE方法在SMOTE的基础上进行了改进，不同于SMOTE中利用所有少数类样本点进行合成采样，而是仅对图中danger类的少数类样本观测点进行合成过采样以改善样本分布。相比于SMOTE，BorderlineSMOTE中少数类合成样本的分布更加合理、紧致，在保证过采样数量的情况下也能避免机器学习模型在训练时过多地学习正样本的信息从而丢失对于负样本的判断能力。

除了上述两种方法外，还有ADASYN等基于SMOTE的过采样方法。根据不同数据集的样本分布特点，不同的过采样方法可能在不同的数据上表现不一，因此探究针对具体的某个不平衡数据集上哪种过采样方法更加适合是很重要的研究目标。鉴于真实信贷业务场景中，在训练数据特征维度高、正负样本极度不平衡的情况下，我们采用上述3种过采样方法进行数据平衡并进行对比。同时，对比未经过采样方法的原始数据训练得到的模型，探究过采样方法对于处理不平衡数据的辅助作用。这种方式既避免了原有训练数据的信息丢失，也能增加模型对正样本的辨识能力，提高分类器精度。

4.2.2 客户端模块

客户端模块主要作用是选用合适的机器学习方法在本地进行训练。考虑到风险控制提前预测该交易是否危险是一个分类问题，我们也选用相应的机器学习模型来学习特征空间到标签空间之间的映射规律：

（1）逻辑回归（Logistic Regression, LR）

逻辑回归本质是将输入空间线性映射到一维空间中，通过逻辑函数：

|  |  |  |
| --- | --- | --- |
|  |  | (4.3) |

将线性回归模型表达式代入后，得到逻辑回归表达式：

|  |  |  |
| --- | --- | --- |
|  |  | (4.4) |

经过LR模型回归后，可以得到借款人的信用分类。LR模型简洁高效，经常被用于各种二分类模型中。

（2）随机森林（Random Forest, RF）

随机森林是一种由机器学习中集成学习（Ensemble Learning）思想中的Bagging策略指导的，由多棵互无关联的决策树（Decision Tree, DT）所构成的一种机器学习算法。它通过随机重复抽取样本集中的样本，按照信息熵变化生成多棵决策树，基于每一棵树的输出结果按照多数投票法（Boyer-Moore Algorithm）进行联合决策，或是加权平均得到分类结果。

（3）梯度提升决策树（Gradient Boosting Decision Tree, GBDT）

梯度提升决策树是一种由多棵决策树构成的迭代算法，与随机森林同属集成学习方法，最初用于回归问题，调整参数设置后也可用于分类问题。在第*k* + 1轮时，每棵决策树学习的是第*k*轮所有决策树的结论和的残差（即负梯度），具有良好的分类效果，被广泛应用于各类数据挖掘竞赛中。

（4）XGBoost（Extreme Gradient Boosting）

XGBoost是近年来各类数据挖掘竞赛中最火热的机器学习算法之一，该模型也可视作为基于决策树的一种梯度提升树。设数据集*D*，在一棵回归树的结果*ft* = 0处，XGBoost希望在第*t*轮的迭代中，以贪心算法思想在该处的泰勒二阶展开去近似它，目标函数近似为：

|  |  |  |
| --- | --- | --- |
|  |  | (4.5) |

其中，分别是*l*的一阶、二阶导数，为正则项。

（5）XGBoostRF

XGBoostRF在XGBoost的基础上使用随机森林算法做作集成，综合了随机森林适配高维度数据、泛化能力强、训练速度快的特点，在xgboost扩展包中以XGBRFClassifier的形式提供接口。

（6）一维卷积神经网络（Convolutional Neural Networks, CNN）

卷积神经网络是指包含卷积计算的深度前馈神经网络（Feedforward Neural Network, FNN），其权值共享、局部连接的特点使得模型对降维后的局部特征较敏感，能较好地学习其中的映射关系。我们在信贷风控系统的客户端尝试采用一维卷积神经网络处理序列数据。

以上6种机器学习模型为我们在客户端的备选机器学习方法，通过实验对比验证其有效性。

4.2.3 中心服务器模块

**（一）安全机制模块**

联邦学习的安全机制除了由“数据不动模型动，数据可用不可见”的本身特点提供外，我们还将差分隐私引入通信算法，作为信贷风控系统安全性的额外保障。由式(2.8)定义了-差分隐私，除了该定义外，还需要介绍*Ln*全局敏感度：

**定义4-1** ***Ln*全局敏感度**[30]对于一个算法，对于任意两个相邻数据集*D*和*D*，*Ln*全局敏感度为：

|  |  |  |
| --- | --- | --- |
|  |  | (4.6) |

全局敏感度与数据集数据无关，只由算法决定。针对具体的某一种*Li* (i = 1, 2, ..., n)敏感度，全局敏感度大小与差分隐私添加的噪声大小成正比：当全局敏感度较大时，算法的输出会因数据集的细微变化而剧烈变化，此时要添加较大的噪声才能保证噪声会作用到数据传输的过程中，否则会因数据集自己的变化掩盖而不明显；当全局敏感度较小时，算法的输出只会由于数据集的细微变化而细微变化，此时只需添加较小的噪声就能实现差分隐私保护。对于数值型的借款人相关数据来说，噪声机制一般选用拉普拉斯机制或高斯机制[56]。这里简单介绍高斯机制，高斯分布的概率分布函数为：

|  |  |  |
| --- | --- | --- |
|  |  | (4.7) |

高斯分布的均值为，方差为，高斯分布噪声机制属于*L*2敏感度。因此添加了高斯噪声机制的联合函数为：

|  |  |  |
| --- | --- | --- |
|  |  | (4.8) |

其中，为真实的映射函数或算法，为添加了高斯噪声机制后的映射函数或算法，为均值为、方差为的高斯分布。具体来说，本地机器学习模型以CNN为例，对于第*k*个客户端，按照式(4.9 - 4.11)进行更新本地模型参数：

|  |  |  |
| --- | --- | --- |
|  |  | (4.9) |
|  |  | (4.10) |
|  |  | (4.11) |

其中，为第*k*个客户端在第*t* + 1轮的模型参数，为本地模型的学习率（步长），为梯度，为损失函数，*B*为本地模型单次训练的所有数据，*C*为梯度裁剪边界值，为第*t*轮全局通信后的全局模型参数。

对于中心服务器，只需要在各客户端进行参数聚合时添加高斯噪声：

|  |  |  |
| --- | --- | --- |
|  |  | (4.12) |

其中，为第*t*轮全局模型参数聚合后的参数结果。

**（二）全局参数聚合模块**

在联邦学习中，全局参数模型聚合一般使用FedAvg[2]方法：

|  |  |
| --- | --- |
| **算法1** **FederatedAveraging**. 共*K*个客户端，每轮参与训练的比例= *C*；*B*为本地训练的batch大小；*E*为本地客户端的迭代轮数；为学习率（步长）；*T*为全局通信轮数. | |
| 1 | 中心服务器执行： |
| 2 | 初始化 |
| 3 | **for** to *T* **do** |
| 4 | *m*max(*CK*, 1) |
| 5 | (从*K*个客户端中随机选择*m*个组成集合) |
| 6 | **for** each client **in parallel** **do** |
| 7 | ClientUpdate(*k*,) |
| 8 | **end for** |
| 9 |  |
| 10 | **end for** |
| 11 | **ClientUpdate(***k*,**)**: **//** 第*k*个客户端的模型更新 |
| 12 | (将本地所有数据以每份*B*的大小划分为多个集合) |
| 13 | **for** each local epoch *i*1 to *E* **do** |
| 14 | **for** batch b**do** |
| 15 | (, b) |
| 16 | **end for** |
| 17 | **end for** |
| 18 | 将参数返给中心服务器 |

在flwr联邦学习框架的FedAvg方法中，客户端*k*的参数在全局共*K*个客户端中的聚合占比为：

|  |  |  |
| --- | --- | --- |
|  |  | (4.13) |

其中，*N*为训练样本数。在本课题所研究的金融场景下的风控系统中，过采样方法平衡正负样本比例后，每个客户端参与训练的本地数据几乎一致，这就导致了原始的依靠训练样本数量来进行加权平均变为了普通的算术平均。同时针对数据的异构特点、训练公平性等问题，有不少学者在FedAvg方法基础上进行改进、完善，如FedAvgM[59]和QFedAvg[60]等。针对金融场景下的交易类别数量不平衡问题，本文提出了一种基于FedAvg的参数聚合方法FedAvgOS（FedAvg for Over Sampling），该方法记录每个本地客户端在本地过采样后，对于每轮迭代的训练数据的Loss平均值，并以该平均值的倒数作为全局参数聚合时的权重。具体步骤如下：

|  |  |
| --- | --- |
| **算法2** **FedAvgOS**. 共*K*个客户端，每轮参与训练的比例= *C*；*B*为本地训练的batch大小；*E*为本地客户端的迭代轮数；为学习率（步长）；*T*为全局通信轮数. | |
| 1 | 中心服务器执行： |
| 2 | 初始化 |
| 3 | **for** to *T* **do** |
| 4 | *m*max(*CK*, 1) |
| 5 | (从*K*个客户端中随机选择*m*个组成集合) |
| 6 | **for** each client **in parallel** **do** |
| 7 | ClientUpdate(*k*,) |
| 8 |  |
| 9 | **end for** |
| 10 | 丢弃数量为*n*drop = 个具有最高Loss值的客户端参数 |
| 11 | **for** each clientin parallel **do** |
| 12 | loss\_reciprocal\_sum |
| 13 | **end for** |
| 14 |  |
| 15 | **end for** |
| 16 | **ClientUpdate(***k*,**)**: **//** 第*k*个客户端的模型更新 |
| 17 | (将本地所有数据以每份B的大小划分为多个集合) |
| 18 | **for** each local epoch *i*1 to *E* **do** |
| 19 | **for** batch b**do** |
| 20 | (, b) |
| 21 | **end for** |
| 22 | **end for** |
| 23 | 将参数返给中心服务器 |

本文提出的FedAvgOS参数聚合方法从各个本地客户端的Loss值出发，考虑到由于本轮过采样的随机性可能会导致对上一轮聚合后的全局模型产生不利影响，因此在全局参数聚合时，丢弃Loss值最高的20%的训练客户端，剩下的客户端以自己的Loss值的倒数在所有客户端的Loss值的倒数和中的占比作为权重，全局参数再加权平均。从理论上，该方法能有效降低过采样随机性带来的影响，使得全局模型参数聚合更加稳定、收敛更加快速。

4.2.4 本节常用符号汇总

**表4-2 3本节符号汇总**

**Table 4-2 Symbol Summary of This Section**

|  |  |
| --- | --- |
| **符号** | **含义** |
|  | 概率分布均值 |
|  | 概率分布标准差 |
|  | 样本差异系数 |
|  | 正则项 |
| *L*n | *L*n全局灵敏度 |
|  | *x*的*n*-范数 |
|  | 某种特定算法或函数映射 |
| *D* | 数据集 |
|  | 高斯分布概率函数 |
|  | 满足均值为，方差为的高斯分布 |
|  | 第*t*轮的全局模型参数 |
|  | 第*k*个客户端在第*t*轮时的本地模型参数 |
|  | 学习率（步长） |
|  | 梯度 |
|  | 损失函数 |
| *B* | 本地模型batch的大小 |
|  | 本地所有数据以每份*B*的大小划分为的集合 |
| *C* | 梯度裁剪边界值 |
|  | 第*t*轮全局参数聚合后的结果 |
| *E* | 本地客户端模型Epoch数量 |
| *T* | 全局通信轮数 |

4.3 实验与结果分析

4.3.1 实验环境

实验环境同开发环境。

4.3.2 实验数据分析

实验数据来自全球知名数据挖掘竞赛平台Kaggle，取自2013年9月中2天时间内的交易数据。样本共计284807条，其种输入空间经过PCA降维后转换为了28维独立的特征数值，加上未经过PCA变换的交易时间Time和交易金额Amount两维数据，输入空间共有30维。输出空间只有1维，名为Class，代表这笔交易是否为欺诈交易，为1表示欺诈交易，为0表示正常交易。下面是整个数据集关于Class的描述：

**表4-3 4数据集信息表**

**Table 4-3 The Information Table of the Data Sets**

|  |  |  |  |
| --- | --- | --- | --- |
| **样本数量** | **正常样本（负样本）** | **欺诈样本（正样本）** | **正负样本比例** |
| **284807** | 284315 | 492 | 1 : 578 |

在数据集中，客户端每次取数据训练时会首先将整个数据集随机打乱并按7 : 3的比例分为训练集和测试集。同时我们关注到原始数据信息中1 : 578极为不平衡的正负样本比例，所以接下来在训练集中，我们对训练集分别按照1 : 1、1 : 50、1 : 100进行过采样，加上原始比例共计4种正负样本比例进行训练。

4.3.3 实验评价指标

在3.3.2节中已经提到过，单一的准确率Accuracy已经不足以衡量分类器的性能，在负样本数量远超正样本数量时，即使将正样本全部预测为负样本，该指标由于其计算公式的定义，也会非常高，所以必须综合其他指标来联合评价信贷风控系统的性能。首先介绍混淆矩阵及几个常用的指标：

**表4-4 5混淆矩阵**

**Table 4-4 Confusion Matrix**

|  |  |  |
| --- | --- | --- |
| **真实标签**  **预测标签** | **正样本** | **负样本** |
| **正样本** | TP | FP |
| **负样本** | FN | TN |

|  |  |  |
| --- | --- | --- |
|  |  | (4.14) |
|  |  | (4.15) |
|  |  | (4.16) |
|  |  | (4.17) |

在机器学习中，我们一般将中的取1，即*F*1值：

|  |  |  |
| --- | --- | --- |
|  |  | (4.18) |

同时，针对信贷风控系统的二分类结果，我们还选用AUC（Area Under the Curve）作为我们的评价指标之一。在二分类模型中，根据不同的既定阈值，每种阈值的设定会得到不同的假正率（FPR，即在所有实际为负样本的样本中，被错误判断为正样本的比例）和真正率（TPR，即在所有实际为正样本的样本中，被正确判断为正样本的比例），以假正率为横坐标，真正率为纵坐标，绘制每个阈值下的(FPR, TPR)形成的曲线被称为ROC（Receiver Operating Characteristic Curve）曲线，曲线下方的面积就是AUC值。

在这些指标中，我们重点关注Recall和AUC值，因为这两者更能在不平衡数据的情况下衡量模型对于少数类样本的识别能力。Recall值衡量的是真实正样本中，系统能够识别正确的比例。在实际的金融场景中，我们更关心风险交易而非正常交易，因为漏判一起风险交易的代价远大于误判一起正常交易的代价[58]。

4.3.4 实验结果分析

**（一）本地客户端非联邦建模实验**

首先，针对非联邦建模，我们选用不同的本地客户端机器学习模型以及过采样方法，探究针对真实金融数据集下的本地模型效果。下面是各模型的基本参数设置：

**表4-5 6本地客户端模型参数设置**

**Table 4-5 The Parameter Settings of the Local Client Models**

|  |  |
| --- | --- |
| **客户端模型** | **基本参数设置** |
| LR | 正则项：l2；停止求解阈值：1e-4；优化算法：lbfgs |
| RF | 决策树个数：100；节点划分标准：gini；决策树深度最大值：不限；抽样策略：有放回；节点最小可分样本数：2；叶节点数：不限 |
| GBDT | 损失函数：对数损失函数deviance；步长：0.1；弱学习器个数：100；抽样策略：不放回；停止求解阈值：1e-4；决策树最大深度：不限 |
| XGBoost | 分类器：树形；树最大深度：6，防止过拟合以及欠拟合；步长：初始0.1，收缩步长策略；正则项：l2 |
| XGBoostRF | 同XGBoost及RF |
| CNN | 网络结构：1维卷积层（输入输出信道：132，卷积大小：31，卷积步长：3）1维池化层（窗口大小：2，步长：1）1维卷积层（输入输出信道：3264，卷积大小：232，卷积步长：2）1维池化层（窗口大小：2，步长：1）全连接层（19264）全连接层（6410）全连接层（102）；优化器：Adam（学习率：1e-4）；损失函数：CrossEntropyLoss()；batchSize：64；epochs：30 |

在这6种模型基础上，将数据集按7 : 3划分训练集、测试集，对训练集进行过采样，固定过采样正负样本比例为1 : 100。1 : 100较符合真实金融场景下的比例，因为有研究称误判100个正常交易的代价大约等于漏判1个风险交易[58]。测试集数据共85443条，其中正样本148条。测试指标有Accuracy、Precision、Recall、F1-score、AUC值，取4位有效数字，衡量指标仅针对原始正样本，而非正样本、负样本的加权平均指标。

**表4-6 7本地客户端模型在不同过采样方法下的结果**

**Table 4-6 The Performance of the Local Client Models with Different Over-Sampling Methods**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Evaluation Index** | | | | | **Sampling Methods** |
| **Accuracy** | **Precision** | **Recall** | **F1** | **AUC** |
| CNN | **0.9995**[[2]](#footnote-1) | **0.8952** | 0.8198 | **0.8558** | 0.9287 | original |
| 0.9992 | 0.7193 | 0.8808 | 0.7921 | 0.9277 | SMOTE |
| 0.9990 | 0.6434 | **0.9128** | 0.7548 | 0.6487 | ADASYN |
| 0.9993 | 0.7851 | 0.8721 | 0.8211 | **0.9345** | Borderline-SMOTE[[3]](#footnote-2) |
| LR | 0.9992 | **0.8780** | 0.6279 | 0.7322 | 0.9807 | original |
| **0.9993** | 0.8047 | 0.8023 | **0.8035** | 0.9839 | SMOTE |
| 0.9992 | 0.7454 | 0.8169 | 0.7795 | **0.9860** | ADASYN |
| 0.9992 | 0.7428 | **0.8227** | 0.7807 | 0.9730 | Borderline-SMOTE |
| RF | **0.9998** | **0.9756** | 0.9302 | **0.9524** | 0.9792 | original |
| **0.9998** | 0.9669 | 0.9331 | 0.9497 | 0.9830 | SMOTE |
| **0.9998** | 0.9525 | 0.9331 | 0.9427 | **0.9872** | ADASYN |
| **0.9998** | 0.9612 | **0.9360** | 0.9485 | 0.9821 | Borderline-SMOTE |
| GBDT | 0.9984 | 0.6977 | 0.1744 | 0.2791 | 0.3125 | original |
| **0.9995** | 0.8462 | 0.8634 | 0.8547 | **0.9881** | SMOTE |
| 0.9994 | 0.8090 | **0.8866** | 0.8460 | 0.9864 | ADASYN |
| **0.9995** | **0.8592** | 0.8692 | **0.8642** | 0.9760 | Borderline-SMOTE |
| XGBoost | **0.9998** | **0.9726** | 0.9273 | 0.9494 | 0.9930 | original |
| **0.9998** | 0.9611 | 0.9331 | 0.9469 | **0.9954** | SMOTE |
| **0.9998** | 0.9446 | **0.9419** | 0.9432 | 0.9950 | ADASYN |
| **0.9998** | 0.9614 | **0.9419** | **0.9515** | **0.9954** | Borderline-SMOTE |
| XGBoostRF | **0.9996** | **0.9470** | 0.8314 | **0.8854** | 0.9772 | original |
| **0.9996** | 0.9065 | 0.8459 | 0.8752 | 0.9820 | SMOTE |
| 0.9995 | 0.8333 | **0.8576** | 0.8453 | **0.9865** | ADASYN |
| 0.9994 | 0.8295 | 0.8488 | 0.8391 | 0.9664 | Borderline-SMOTE |

由表4-6可知，不同的过采样方法在不同的模型中的表现差异较大。其中，ADASYN和Borderline-SMOTE方法分别在3种不同的本地模型上都取得了最佳的Recall值，证明其对正样本的随机合成具有优越性；但ADASYN方法合成的正样本特征太强，反而导致其Precision值在6种模型上都表现不佳。同时注意到，RF、XGBoost等基于Bagging或Boosting策略的模型表现优异，且原始正负样本的比例对模型效果影响不大，如果在一般金融场景的非联邦建模中可以采纳。观察可知，按照1 : 100的比例将正负样本平衡，在Recall值上相对于原始数据1 : 578的正负样本比例来说几乎都有较大幅度的提升。也就是说，提高训练集中的正样本数量，可以确定地提高对真样本的辨别比能力。但与此同时，注意到Precision和F1-score指标相对于原始数据却并未全面提升，甚至还有下降，推测是由于对正样本的过度采样导致学习后的模型会过多地学习正样本的数据特征从而在测试集中将负样本错判为正样本。从以上结果中观察发现CNN模型在经过采样后具有较好的正样本辨别能力，同时由于神经网络其特殊性，我们可以很方便地获取其每一层网络的非超参数并以tensor的形式进行传输共享，所以我们确定CNN作为之后联邦建模实验的本地客户端模型。

**（二）联邦建模实验**

（1）过采样比例实验

下面开始进行联邦建模实验。为进一步验证过采样比例猜想，我们使用在6种模型上综合表现较好的Borderline-SMOTE方法，所有备选客户端（共10个）全部参与训练，按照1 : 1、1 : 50、1 : 100、以及原始比例共4种比例进行联邦建模实验，观察并记录实验结果。

**表4-7 8本地客户端模型在不同过采样比例下的结果**

**Table 4-7 The Performance of the Local Client Model with Different Over-Sampling Ratios**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Evaluation Index** | | | | |  | **Sampling Ratio** |
| **Accuracy** | **Precision** | **Recall** | **F1** | **AUC** | **Times/s** |
| CNN | 0.9994 | 0.7963 | **0.8750** | 0.8338 | 0.9432 | 6319 | 1 : 1 |
| **0.9996** | 0.9015 | 0.8663 | 0.8759 | 0.9403 | 3710 | 1 : 50 |
| **0.9996** | 0.9049 | 0.8567 | 0.8806 | **0.9462** | 3681 | 1 : 100 |
| **0.9996** | **0.9502** | 0.8314 | **0.8868** | 0.9157 | **3453** | original |

|  |  |
| --- | --- |
| (a) Loss | (b) Accuracy |
| (c) Precision | (d) Recall |
| (e) AUC | (f) F1-score |

**图4-3 8不同过采样比例下的联邦建模指标对比**

**Figure 4-3 The Comparison of Indexes between Different Over-Sampling Ratios**

由表4-7及图4-3可知，不同的过采样比例下的Recall值和AUC值均高于未经过采样的联邦学习的对应指标。在联邦建模中，随着过采样比例的提高，Precision值依次降低，Recall值逐渐升高。我们观察到在1 : 1过采样下系统有着最大的Recall值，但整个联邦训练的耗时几乎是其他比例的2倍。而1 : 100下的表现与1 : 50下的表现接近，且耗时更低，因此我们取1 : 100的过采样比例进行后续的联邦建模实验。

（2）联邦学习本地客户端训练数量实验

在联邦建模与非联邦建模的对比中，我们在之后的实验中将参与联邦学习的客户端数量设置为N = 10，对每次参与联邦学习参数上传融合的客户端数量和新模型测试的客户端数量，有：

|  |  |  |
| --- | --- | --- |
|  | *Nrfit* = *nfit* | (4.19) |
|  | *N**reval* = *neval* | (4.20) |

其中，*rfit*为参与联邦训练的本地客户端比例，*reval*为参与联邦测试的本地客户端比例。我们分别设置*rfit* = 0.2、*rfit* = 0.5、*rfit* = 0.8，同时令*reval* = 0.5，结合非联邦集中式训练参数，观察相关指标随通信轮数的变化。每个客户端训练时在本地迭代5次，全局通信30次，记录结果并与非联邦建模的集中式训练（Centralized）对比。

**图4-4 不同客户端数量下的指标对比**

**Figure 4-4 The Comparison of Indexes between Different Numbers of Training Clients**

|  |  |
| --- | --- |
| (a) Loss | (b) Accuracy |
| (c) Precision | (d) Recall |
| (e) AUC | (f) F1-score |

**图4-4（续）9不同客户端数量下的指标对比**

**Figure 4-4(continued) The Comparison of Indexes between Different Numbers of Training Clients**

由图4-4可知，在6项指标上，联邦建模的平均表现整体来看都要优于非联邦建模，尤其提高了单一客户端上的Precision值，证明联邦学习可以提高风控模型防止误判的能力。联邦建模在通信轮数的后半部分，虽然平均表现要稳定好于集中式训练，但联邦学习的各类指标，尤其是Loss值和Precision值会在一定范围内产生较大波动，收敛效果一般，推测可能是以下2个原因导致：（1）过采样方法合成的正样本具有随机性，多个客户端每轮都是用不同的、随机划分的、随机过采样后的训练数据，使得其样本分布方差要大于单个客户端的集中式训练，导致模型收敛较慢；（2）带有差分隐私机制的联邦建模每轮会对上传至中心服务器的参数添加噪声并且在客户端会进行梯度裁剪，随着模型逐渐收敛，差分隐私的代价逐渐变大。同时在3种不同的*rfit*下，随着*rfit*的提高，全局模型在通信轮数的后半段更加稳定，尤其是在Loss值和AUC值上体现明显。然而我们观察到，经过过采样合成的正样本仍具有一定的随机性，导致联邦学习进行全局参数聚合时，会因为某些客户端在第*t*轮时过采样后的样本质量较低导致全局模型在第*t* + 1轮的表现不佳。为此，提出了一种针对联邦学习客户端过采样情况下的参数聚合方法FedAvgOS，下面是该方法的实验部分。

**表4-8 9不同参数聚合方法下的实验数据对比**

**Table 4-8 The Comparison between Different Parametric Polymerization Methods**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Method** | **Evaluation Index** | | | | | | **Statistics** |
| **Loss** | **Accuracy** | **Precision** | **Recall** | **F1** | **AUC** |
| Centralized | 0.5790 | 0.9991 | 0.6990 | 0.8580 | 0.7680 | 0.9302 | Mean |
| 0.1237 | 0.0003 | 0.0734 | 0.0056 | 0.0478 | 0.0022 | Standard Deviation |
| *rfit* = 0.5 | 0.4032 | 0.9994 | 0.8115 | 0.8675 | 0.8366 | **0.9412**[[4]](#footnote-3) | Mean |
| FedAvg[2] | 0.0645 | 0.0001 | 0.0514 | 0.0036 | 0.0281 | 0.0028 | Standard Deviation |
| *rfit* = 0.5 | 0.4460 | 0.9994 | 0.7878 | **0.8687** | 0.8253 | 0.9364 | Mean |
| FedAvgM[59] | 0.0623 | 0.0001 | **0.0372** | 0.0028 | 0.0219 | 0.0148 | Standard Deviation |
| *rfit* = 0.5 | **0.3850** | **0.9995** | **0.8495** | 0.8558 | **0.8514** | 0.9324 | Mean |
| **FedAvgOS** | **0.0484** | **0.0001** | 0.0420 | **0.0024** | **0.0216** | **0.0013** | Standard Deviation |
| *rfit* = 0.8 | 0.3980 | 0.9994 | 0.8227 | 0.8645 | 0.8409 | 0.9422 | Mean |
| FedAvg | **0.0509** | 0.0001 | 0.0497 | 0.0040 | 0.0258 | 0.0035 | Standard Deviation |
| *rfit* = 0.8 | 0.3955 | 0.9994 | 0.8310 | 0.8605 | 0.8434 | 0.9390 | Mean |
| FedAvgM | 0.0660 | 0.0002 | 0.0627 | 0.0034 | 0.0327 | **0.0026** | Standard Deviation |
| *rfit* = 0.8 | **0.3894** | **0.9995** | **0.8338** | **0.8650** | **0.8477** | **0.9441** | Mean |
| **FedAvgOS** | 0.0559 | **0.0001** | **0.0460** | **0.0019** | **0.0253** | 0.0063 | Standard Deviation |

**图4-5 FedAvgOS方法实验结果**

**Figure 4-5 Results of FedAvgOS Experiments**

|  |  |
| --- | --- |
| (a) Loss with *rfit* = 0.5 | (b) Accuracy with *rfit* = 0.5 |
| (c) Precision with *rfit* = 0.5 | (d) F1-score with *rfit* = 0.5 |
| (e) Accuracy with *rfit* = 0.8 | (f) Recall with *rfit* = 0.8 |
| (g) F1-score with *rfit* = 0.8 | (h) AUC with *rfit* = 0.8 |

**图4-5（续）10FedAvgOS方法实验结果**

**Figure 4-5(continued) Results of FedAvgOS Experiments**

表4-8显示了在非联邦建模和联邦建模下不同的训练比例*rfit*下，各评价指标在固定通信轮数E = 30下的平均值和标准差。我们希望在中心服务器的参数聚合中，全局参数的聚合能尽可能减小部分因为过采样质量较低或过采样后样本分布方差过大的客户端参数影响，所以以平均值和标准差来分别衡量全局参数聚合的性能优劣与稳定性的大小。表中数据显示，在相同的训练比例*rfit*下，FedAvgOS的绝大部分指标的平均值都要大于FedAvg及改进的FedAvgM[59]方法，标准差都要小于FedAvg及FedAvgM方法，证明了我们提出的FedAvgOS在本地客户端过采样情况下的参数聚合都具有良好的性能。

图4-5 (a)(b)(c)(d) 显示了在*rfit* = 0.5情况下，FedAvgOS方法在Loss、Accuracy、Precision以及F1-score共4项指标随通信轮数变化的变化。经过FedAvgOS方法先丢弃一部分在本地训练集上表现较差的客户端参数，剩余客户端按照Loss值的倒数作为权重进行加权平均。这种方法使得全局模型表现在通信轮数的中后半段表现更加稳定，尤其是在Loss值上表现明显优于未对过采样场景下优化的FedAvg及FedAvgM方法。

图4-5 (e)(f)(g)(h) 显示了在*rfit* = 0.8情况下，FedAvgOS方法在Accuracy、Recall、AUC以及F1-score共4项指标随通信轮数变化的变化。相较于*rfit* = 0.5情况下，FedAvgOS方法表现更加良好，波动程度减小，充分说明了FedAvgOS方法的优越性。

与非联邦建模相比，结合联邦学习技术尤其是本文提出的FedAvgOS参数聚合方法的信贷风控模型，在对正样本进行过采样的条件设置下，具有更好的性能，切实提高了信贷风控模型对于风险交易的判别预警能力以及对于非风险交易的防误判能力。

4.4 本章小结

本章从基于联邦学习的信贷风控系统实际构建出发，对第3章设计的各模块中的重点步骤、方法等以表格或公式的形式作了详细介绍。接着进入实验部分，首先分析实验环境、介绍实验指标，接着在数据上观察不同过采样方法以及不同的过采样比例对实验结果的影响，遴选出表现较好的本地客户端模型以及采样方法然后进行联邦建模，验证了联邦学习对于金融信贷风控的适应性和有效性，说明联邦学习技术用于金融领域，特别是信用风险控制领域是切实可行的。在过采样的基础上，提出了FedAvgOS参数聚合算法，相比于基线FedAvg算法及FedAvgM算法能够一定程度上缓解在正负样本数量不平衡的情况下，过采样方法带来的不确定性，从而加速全局模型收敛速度。

**第5章 总结与展望**

5.1 课题工作总结

本文从金融科技的蓬勃发展背景引出传统金融行业业务与联邦学习技术各自的背景，并分析两种不同领域内各自发展的特点、难点，提出将金融行业内的信用贷款风险控制业务使用联邦学习建模，以解决数据孤岛、数据隐私安全、数据倾斜以及数据异构等问题。在广泛阅读相关文献、概览行业前沿进展后，分析了联邦学习用于信贷风控领域的可行性，接着设计了基于联邦学习的信贷风控系统，并通过一系列实验进行了样本极度倾斜情况下的过采样方法及比例探究，以及联邦学习中对于过采样方法的参数融合方法改进，使得在不同客户端面对不完全一样的随机过采样数据时，增强了联邦模型的鲁棒性，从而不至于因某些客户端的过采样正样本质量相对较低导致联邦全局模型的精度降低。最后本文的研究工作可从以下几个方向总结：

（1）概述了金融行业传统信用贷款风险控制的背景，分析了联邦学习技术用于信贷风控领域的可行性。金融行业的数据高度敏感、高度隐私，各家金融机构的私有数据几乎不可能融合，联邦学习为多家金融机构进行联合风控建模提供了全新的思路。

（2）详细对结合机器学习方法的信贷风控业务研究进行梳理，并从联邦学习的基本概念与分类出发，详细介绍了其结构、研究现状以及未来应用背景。

（3）设计了针对一般金融场景的正负样本数量极度倾斜的情况下，通过过采样方法先将样本平衡至既定比例，再通过联邦学习本地客户端训练——中心服务器聚合的训练模型，经过实验验证，联邦学习完全可以用于金融行业中的风控建模，并实现较一般本地集中式训练性能更为优越的联邦全局模型。

（4）针对一般过采样方法其少数类样本随机合成的特点，设计了一种增强模型参数聚合鲁棒性的参数聚合方法FedAvgOS，使得多个客户端模型经过本地训练随后进行参数聚合时，不受单个或部分的因过采样样本分布不同、分布偏离的客户端影响，从而导致全局模型参数平均后精度下降。实验说明使用FedAvgOS参数聚合方法的联邦全局模型在迭代收敛时的损失波动以及绝大部分指标优于基线方法。

5.2 未来工作展望

本课题所提出的基于联邦学习的信贷风控系统可一定程度上提升银行等金融机构对信用贷款的风险防范能力。但由于时间因素等，本课题的研究工作还存在可持续改进之处，具体如下：

（1）信用卡交易欺诈是信贷风控的重要一环，但整个信贷风控的流程不仅由欺诈交易检测构成，还包括贷前、贷中、贷后各步骤中的各个环节。因此除了欺诈交易风险控制外，未来可在现有联邦学习的研究基础上，继续完善整个信贷风控体系的联邦建模，切实解决或缓解行业痛点、难点。

（2）在本地客户端建模时发现，虽然如CNN等的神经网络模型具有效果好、精度高等特点，但相较于数据挖掘中的最新基于信息熵理论、Bagging或Boosting技术的树模型（随机森林、XGBoost等），仍然具有可解释性较差、训练速度较慢等特点，而针对现实场景中金融数据的强特征性、高并发性等特点，仍需要提升本地客户端模型的可解释性以及训练速度。

（3）本课题所选数据来源于现实金融场景经过PCA方法降维并标准化后的欺诈交易数据，相较于现实金融场景的第一手原始数据，可能会丢失部分信息从而影响本模型对于现实第一手数据的泛化能力。未来可以通过购买等方式拿到不同金融机构的真实异构数据，再进行联邦建模研究。

（4）本文针对联邦建模安全性所结合的差分隐私保护方法，由于技术有限，并未经过实际的安全性验证测试，无法确保其对于半可信、不可信或恶意的客户端或中心服务器的防范能力。未来可在联邦建模中的安全性提升、客户端激励与筛选等方面继续开展研究，从而提升联邦模型的安全性以及性能。

**参考文献**

1. 海外网. 法国政府对谷歌罚款1.5亿欧元 称其违反隐私规则[EB/OL].(2022-01-06)[2022-04-20]. https://baijiahao.baidu.com/s?id=1721205135347754106&wfr=spider&for=pc.
2. McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. New York, NY: PMLR, 2017: 1273-1282.
3. 杨强, 黄安埠, 刘洋, 陈天健. 联邦学习实战[M]. 第1版. 北京: 电子工业出版社, 2021: 6-6.
4. 李航. 统计学习方法[M]. 第2版. 北京: 清华大学出版社, 2012: 3-15.
5. Durand D. Risk elements in consumer installment financing[M]. National Bureau of Economic Research, New York, 1941: 60-72.
6. Orgler Y E. A credit scoring model for commercial loans[J]. Journal of Money, Credit and Banking, 1970, 2(4): 435-445.
7. Wiginton J C. A note on the comparison of logit and discriminant models of consumer credit behavior[J]. Journal of Financial and Quantitative Analysis, 1980, 15(3): 757-770.
8. Grablowsky B J, Talley W K. Probit and discriminant functions for classifying credit applicants-a comparison[J]. Journal of Economics and Business, 1981, 33(3): 254-261.
9. Ziari H A, Leatham D J, Ellinger P N. Development of statistical discriminant mathematical programming model via resampling estimation techniques[J]. American Journal of Agricultural Economics, 1997, 79(4): 1352-1362.
10. Bian Y, Cheng M, Yang C, et al. Financial fraud detection: a new ensemble learning approach for imbalanced data[C]//Proceedings of the 20th Pacific Asia Conference on Information Systems. Association for Information Systems, 2016: 315.
11. Odom M D, Sharda R. A neural network model for bankruptcy prediction[C]//Proceedings of the 1990 International Joint Conference on Neural Networks. Piscataway, NJ: IEEE, 1990: 163-168.
12. Luo C, Wu D, Wu D. A deep learning approach for credit scoring using credit default swaps[J]. Engineering Applications of Artificial Intelligence, 2017, 65: 465-470.
13. 邹雅莹. 基于k-means算法的马田系统研究及其在个人信用评价中的应用[D]. 南京理工大学, 2014.
14. Chougule P, Thakare A D, Kale P, et al. Genetic k-means algorithm for credit card fraud detection[J]. International Journal of Computer Science and Information Technologies, 2015, 6(2): 1724-1727.
15. Tran K, Duong T, Ho Q. Credit scoring model: A combination of genetic programming and deep learning[C]//Proceedings of the 2016 Future Technologies Conference. Piscataway, NJ: IEEE, 2016: 145-149.
16. Zaslavsky V, Strizhak A. Credit card fraud detection using self-organizing maps[J]. Information and Security, 2006, 18: 48.
17. Yang Q, Liu Y, Chen T, et al. Federated machine learning: Concept and applications[J]. ACM Transactions on Intelligent Systems and Technology, 2019, 10(2): 1-19.
18. Hard A, Rao K, Mathews R, et al. Federated learning for mobile keyboard prediction[J]. arXiv preprint arXiv: 1811.03604, 2018.
19. Feng S, Yu H. Multi-participant multi-class vertical federated learning[J]. arXiv preprint arXiv: 2001.11154, 2020.
20. Konečný J, McMahan H B, Ramage D, et al. Federated optimization: Distributed machine learning for on-device intelligence[J]. arXiv preprint arXiv: 1610.02527, 2016.
21. Xie C, Koyejo S, Gupta I. Asynchronous federated optimization[J]. arXiv preprint arXiv: 1903.03934, 2019.
22. Chen J, Pan X, Monga R, et al. Revisiting distributed synchronous SGD[J]. arXiv preprint arXiv: 1604.00981, 2016.
23. Sattler F, Wiedemann S, Müller K R, et al. Robust and communication-efficient federated learning from non-iid data[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(9): 3400-3413.
24. Wang H, Yurochkin M, Sun Y, et al. Federated learning with matched averaging[J]. arXiv preprint arXiv: 2002.06440, 2020.
25. Reddi S, Charles Z, Zaheer M, et al. Adaptive federated optimization[J]. arXiv preprint arXiv: 2003.00295, 2020.
26. Konečný J, McMahan H B, Yu F X, et al. Federated learning: Strategies for improving communication efficiency[J]. arXiv preprint arXiv: 1610.05492, 2016.
27. Caldas S, Konečny J, McMahan H B, et al. Expanding the reach of federated learning by reducing client resource requirements[J]. arXiv preprint arXiv: 1812.07210, 2018.
28. He L, Bian A, Jaggi M. Cola: Decentralized linear learning[J]. Advances in Neural Information Processing Systems, 2018, 31.
29. Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms[J]. Foundations of Secure Computation, 1978, 4(11): 169-180.
30. Dwork C. Differential privacy: A survey of results[C]//Proceedings of the International Conference on Theory and Applications of Models of Computation. Berlin, Heidelberg: Springer, 2008: 1-19.
31. 李少波, 杨磊, 李传江, 张安思, 罗瑞士. 联邦学习概述：技术、应用及未来[J/OL]. 计算机集成制造系统: 1-29[2022-04-24]. http://kns.cnki.net/kcms/detail/11.5946.TP.20210831.1414.006.html.
32. Yao A C. Protocols for secure computations[C]//Proceedings of the 23rd Annual Symposium on Foundations of Computer Science. Piscataway, NJ: IEEE, 1982: 160-164.
33. Yao A C C. How to generate and exchange secrets[C]// /Proceedings of the 27th Annual Symposium on Foundations of Computer Science. Piscataway, NJ: IEEE, 1986: 162-167.
34. Goldreich O, Micali S, Wigderson A. How to play any mental game, or a completeness theorem for protocols with honest majority[M]//Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali. 2019: 307-328.
35. 杨强, 黄安埠, 刘洋, 陈天健. 联邦学习实战[M]. 第1版. 北京: 电子工业出版社, 2021: 26-29.
36. Zhu L, Liu Z, Han S. Deep leakage from gradients[J]. Advances in Neural Information Processing Systems, 2019, 32.
37. Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures[C]//Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. New York, NY: ACM, 2015: 1322-1333.
38. Melis L, Song C, De Cristofaro E, et al. Inference attacks against collaborative learning[J]. arXiv preprint arXiv: 1805.04049, 2018, 13.
39. Wang H, Kaplan Z, Niu D, et al. Optimizing federated learning on non-iid data with reinforcement learning[C]//Proceedings of the 39th IEEE Conference on Computer Communications. Piscataway, NJ: IEEE, 2020: 1698-1707.
40. Mills J, Hu J, Min G. Communication-efficient federated learning for wireless edge intelligence in IoT[J]. IEEE Internet of Things Journal, 2019, 7(7): 5986-5994.
41. Li T, Sahu A K, Talwalkar A, et al. Federated learning: Challenges, methods, and future directions[J]. IEEE Signal Processing Magazine, 2020, 37(3): 50-60.
42. Hanzely F, Richtárik P. Federated learning of a mixture of global and local models[J]. arXiv preprint arXiv: 2002.05516, 2020.
43. Wang X, Han Y, Wang C, et al. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning[J]. IEEE Network, 2019, 33(5): 156-165.
44. 李健萌. 基于联邦学习的大数据风险控制技术研究与应用[D]. 浙江大学, 2020.
45. 杨强, 黄安埠, 刘洋, 陈天健. 联邦学习实战[M]. 第1版. 北京: 电子工业出版社, 2021: 103-105.
46. Maes S, Tuyls K, Vanschoenwinkel B, et al. Credit card fraud detection using Bayesian and neural networks[C]//Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies. 2002, 7.
47. Fan W, Stolfo S J, Zhang J, et al. AdaCost: Misclassification Cost-Sensitive Boosting[C]//Proceedings of the 16th International Conference on Machine Learning. New York, NY: PMLR, 1999: 97-105.
48. Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research, 2002, 16: 321-357.
49. Han H, Wang W Y, Mao B H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning[C]//Proceedings of the 2005 International Conference on Intelligent Computing. Berlin, Heidelberg: Springer, 2005: 878-887.
50. He H, Bai Y, Garcia E A, et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning[C]//Proceedings of the 2008 IEEE International Joint Conference on Neural Networks. Piscataway, NJ: IEEE, 2008: 1322-1328.
51. Batista G E, Prati R C, Monard M C. A study of the behavior of several methods for balancing machine learning training data[J]. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 20-29.
52. Batista G E, Bazzan A L C, Monard M C. Balancing Training Data for Automated Annotation of Keywords: a Case Study[C]//Proceedings of the 2003 Brazilian Workshop on Bioinformatics. 2003: 10-18.
53. Beutel D J, Topal T, Mathur A, et al. Flower: A friendly federated learning research framework[J]. arXiv preprint arXiv: 2007.14390, 2020.
54. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python[J]. Journal of Machine Learning Research, 2011, 12: 2825-2830.
55. Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library[J]. Advances in Neural Information Processing Systems, 2019, 32.
56. Dwork C, McSherry F, Nissim K, et al. Calibrating noise to sensitivity in private data analysis[J]. Journal of Privacy and Confidentiality, 2016, 7(3): 17-51.
57. Dwork C, Roth A. The algorithmic foundations of differential privacy[J]. Found. Trends Theor. Comput. Sci., 2014, 9(3-4): 211-407.
58. 阳文斯. 基于联邦学习的信用卡欺诈检测系统研究[D]. 中国科学院大学(中国科学院深圳先进技术研究院), 2020.
59. Hsu T M H, Qi H, Brown M. Measuring the effects of non-identical data distribution for federated visual classification[J]. arXiv preprint arXiv:1909.06335, 2019.
60. Li T, Sanjabi M, Beirami A, et al. Fair resource allocation in federated learning[J]. arXiv preprint arXiv:1905.10497, 2019.

**声明**

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

学位论文作者（签名）\_\_\_\_\_\_\_\_\_\_

论文指导教师（签名）\_\_\_\_\_\_\_\_\_\_

2022年5月20日

**致谢**

至此，毕业设计已基本完成，我的四年本科生活也划上了一个圆满的句号。

首先就毕业设计而言，我要首先感谢我的导师段磊教授。段老师为我提供了毕业设计的方向与思路，并且在整个过程中，一直细致严谨、认真负责地为我提供指导，让我第一次完成了较为完整的科研训练。其次要感谢实验室的各位师兄师姐，在大四学年进入实验室以来，我一直跟随他们学习科研相关知识，并且有幸能参与到实验室的科研项目及师兄的论文写作中，这也为我最终完成一份较高质量的毕业设计奠定了基础。另外还要感谢实验室提供的便利资源，这是我完成这份毕业设计的物质基础。

回首大学四年，是各位同学、朋友的陪伴，才让我一直走到今天。感谢何帅、伊瑞光等室友一路上的陪伴与帮助，和你们在一起的时光非常快乐。感谢学院素拓部的学长与同学，从你们身上我学到了很多知识与技能。感谢四年中我在参加项目、比赛时遇到的其他学院的同学、朋友，也正是四川大学海纳百川、有容乃大的品质，给不同学院、不同专业的同学提供了这样一个广阔的平台，使我们互相学习进步。感谢叶童等高中同学、大学校友的陪伴与支持，与你们交流、倾诉总能让我有所收获。同时，再次感谢实验室的老师、各位师兄师姐及同学，我也即将进入实验室开启我的硕士研究生生活，希望接下来大家一起努力进步。

我要向我的父母表示感谢，你们细致的关心与无条件的支持是我坚实的后盾，支持我一路走来直至本科毕业。我还要向我的母校四川大学表示感谢，我即将继续在四川大学开启又一个三年，川大人的标签已深深烙印在我的心里。

最后，感谢一路上帮助过自己的老师、朋友，感谢自己一路以来的努力与坚持。

1. https://baike.baidu.com/item/%E5%B7%B4%E5%A1%9E%E5%B0%94%E5%8D%8F%E8%AE%AE/2375654?fr=aladdin [↑](#footnote-ref-0)
2. 黑体数字指对于同种本地客户端模型不同过采样方法在同种指标下的最佳值。 [↑](#footnote-ref-1)
3. 此处指只针对少数类样本进行过采样的BorderlineSMOTE-1算法。 [↑](#footnote-ref-2)
4. 黑体数字表示在相同训练比例下，三种参数聚合方法在固定通信轮数下分别对于平均值（Mean）和标准差（Standard Deviation）下更佳的表现。 [↑](#footnote-ref-3)