文章编号:1671-9352(2026)03-0029-15 DOI:10.6040/j.issn.1671-9352.9.2025.004

恶意被动方场景下的纵向联邦学习安全加权聚合

张政胤[1,2,3],王玲玲[1,2*],黄梅[1,2],张玉兴[1,2],宋佼蓉[1,2]
(1.青岛科技大学信息科学技术学院, 山东 青岛 266042; 2.山东省深海装备智联网重点实验室, 山东 青岛 266042; 3.烟台城市科技职业学院, 山东 烟台 265500)

摘要:针对纵向联邦学习中的不可信参与方发动数据投毒攻击阻碍模型训练,以及半诚实参与方发动隐私推理攻击窃取其他参与方私有数据的问题,提出了一种恶意被动方场景下的纵向联邦学习安全加权聚合方案。首先,设计效用评估算法抵御数据投毒攻击,通过计算最大容忍距离过滤有毒样本所对应的嵌入向量。然后,提出自适应权重计算算法,确保在长尾数据场景下依然能够有效抵御数据投毒攻击并保持模型的高收敛率和准确率。最后,利用掩蔽机制和对称同态加密算法保护嵌入向量隐私,抵御隐私推理攻击。理论分析和仿真结果表明本方案具有较好的计算效率和模型性能,能有效抵御隐私推理攻击和数据投毒攻击,与最新相关工作相比模型准确率提高约 5%~10%。

关键词:纵向联邦学习;数据投毒攻击;隐私推理攻击;长尾数据
中图分类号:TP309.2 **文献标志码**:A
引用格式:张政胤,王玲玲,黄梅,等. 恶意被动方场景下的纵向联邦学习安全加权聚合[J]. 山东大学学报(理学版),2026,61(3):29-43.

Secure weighted aggregation for VFL with malicious passive parties

ZHANG Zhengyin[1,2,3], WANG Lingling[1,2*], HUANG Mei[1,2], ZHANG Yuxing[1,2], SONG Jiaorong[1,2]
(1. School of Information Science and Technology, Qingdao 266042, Shandong, China; 2. Qingdao University of Science and Technology, Qingdao 266042, Shandong, China; 3. Yantai City College of Science and Technology, Yantai 265500, Shandong, China)

Abstract: Considering the problem that untrustworthy participants in vertical federated learning launch data poisoning attacks to hinder model training, and that semi-honest participants launch inference attacks to steal privacy information of other participants, a securely weighted aggregation scheme for vertical federated learning with malicious passive parties is proposed. First, a utility evaluation algorithm is combined to defend against data poisoning attacks, and the maximum tolerance distance is designed to filter the poisoned embedding vectors; Second, an adaptive weight calculation algorithm is designed to ensure that the model can still effectively resist data poisoning attacks and maintain high convergence rate and accuracy in long-tailed data scenarios. Finally, the masking mechanism and symmetric homomorphic encryption algorithm are utilized to protect the privacy of embedding vectors against privacy inference attacks. Theoretical analysis and simulation results show that the proposed protocols has better computational efficiency and model performance, can effectively resist privacy inference attacks and data poisoning attacks, and improves the model accuracy by about 5%-10% compared with the latest related work.

Key words: vertical federated learning; data poisoning attacks; privacy inference attack; long-tail data

0 引言

联邦学习(federated learning, FL)[1]作为一种保护隐私的分布式机器学习范式,允许参与方之间交换训练的中间结果而不是原始数据,从而协作训练机器学习模型。根据不同的数据划分方式,联邦学习可以被分为横向联邦学习(horizontal federated learning, HFL)和纵向联邦学习(vertical federated learning, VFL)。在

收稿日期:2025-01-24; **网络出版时间**:2026-01-19
基金项目:国家自然科学基金资助项目(61802217); 山东省自然科学基金资助项目(ZR2023MF082); 青岛科技计划重点研发项目(22-3-4-xxgg-10-gx); 青岛市自然科学基金原创探索项目(23-2-1-164-zyyd-jch)
第一作者:张政胤(2000—),男,硕士研究生,研究方向为联邦学习隐私保护. E-mail:ZhangZhengyin@ mails.qust.edu.cnn
*通信作者:王玲玲(1982—),女,副教授,博士,研究方向为应用密码学及联邦学习隐私安全. E-mail:wanglingling@ qust.edu.cn

纵向联邦学习中,参与方数据集共享相同的样本空间,但持有不同的特征空间,通过学习来自各参与方相同样本的不同特征来提高模型性能和泛化能力。早期纵向联邦学习工作主要集中在训练简单的机器学习模型,如逻辑回归和决策树。近来,基于模型拆分的纵向联邦学习(split vertical federated learning, SVFL)[2]可以通过融合拆分学习(split learning, SL)达到训练更加复杂模型(如多层感知机和卷积神经网络)的目的。

基于模型拆分的纵向联邦学习,神经网络被拆分为底层模型和顶层模型,被动方作为数据特征的持有者拥有底层模型,主动方作为数据标签的持有者拥有顶层模型。被动方将本地特征映射为嵌入向量并将其发送给主动方,主动方聚合嵌入向量并输入到顶层模型同时计算训练损失。嵌入向量的正确性直接影响主动方的聚合结果,进而影响模型收敛。然而,在复杂开放的网络环境中,不可信的被动方可能发动数据投毒攻击生成错误的嵌入向量并提交给主动方聚合;半诚实的主动方可能会对被动方的私有数据感兴趣从而发动隐私推理攻击。因此,在恶意被动方场景下研究嵌入向量的安全聚合是非常有必要的。

具体来说,恶意被动方可能会在训练数据中掺入随机生成的有毒样本,发动数据投毒攻击破坏和阻碍模型训练过程,导致模型无法正常收敛。这不仅降低了模型的准确性,延长了训练时间,还增加计算成本。VFL 中的数据投毒攻击难以使用传统基于相似度的方法进行检测和抵御,因为有毒嵌入向量在稀疏性和多样性上与良性嵌入向量几乎没有差别。尤其是在长尾数据场景下,这类攻击的检测难度更大。长尾数据场景中,样本分布具有明显的不均衡性,少数类别的样本占据了数据集的大部分,有毒数据可能被误认为是正常的尾部数据,所以使得数据投毒攻击更具隐蔽性。这也导致现有方案在长尾数据场景下难以达到较高的模型准确率。

在开放的 VFL 中,隐私泄露风险无处不在,半诚实的主动方可能会发动隐私推理攻击[3-4],通过分析嵌入向量,利用已知的模型结构和参数,尝试重构被动方的私有数据。这类攻击严重威胁到数据隐私,破坏联邦学习的协作基础。同时,在加权聚合需求下,权重通常由主动方计算,计算过程及需要用到的参数可能会泄露被动方隐私。例如,在基于效用评估的权重计算方案[5-6]中,效用评估算法大都基于排一法,被排除的被动方相关信息会直接暴露给主动方,因此很难在确保隐私的前提下对被动方发送的嵌入向量进行效用评估和权重计算。

为了应对上述挑战,本文提出了一种恶意被动方场景下的纵向联邦学习安全加权聚合方案,旨在有效抵御恶意被动方发动的数据投毒攻击,尤其在长尾数据场景中本文所提方案也能达到较好的数据投毒攻击抵御效果。同时,本文通过效用评估结果计算最大容忍距离来过滤掉恶意被动方发送的有毒嵌入向量,同时根据效用评估结果和特征数量为嵌入向量设计聚合权重,在长尾数据场景下达到较高的模型收敛率和准确率。此外,本文利用掩蔽机制[7]和对称同态加密算法(symmetric homomorphic encryption, SHE)[8]保护嵌入向量的隐私,防止主动方和服务器通过推理嵌入向量来获得被动方的隐私数据。

综上,本文的主要工作如下:

(1) 提出了一个效用评估和权重计算协议,通过计算最大容忍距离过滤有毒数据,并根据效用评估结果设计自适应权重计算算法,确保模型在长尾数据场景下的高收敛率和高准确率。

(2) 设计了一个隐私保护的损失计算协议,允许多个主动方在不得到嵌入向量明文的前提下协作完成顶层模型的推理过程,从而抵御主动方的隐私推理攻击。并通过严格的隐私性分析,证明了该协议具有隐私保护特性。

(3) 对所提协议进行了全面的仿真实验。实验结果表明,在普通场景和长尾数据场景下均能够有效过滤出有毒数据所生成的嵌入向量,与最新相关工作相比准确率提高约 5%~10%。

1 相关工作

1.1 数据投毒攻击下的鲁棒聚合

鲁棒聚合的相关研究多出现在 HFL 中。Shen 等[9]提出对局部更新进行聚类并剔除异常值,以确保训练数据的真实性。Fung 等[10]通过余弦相似度测量梯度的角距离来解决基于 sybil 的数据集中毒问题。Andreina等[11]将反馈循环纳入 FL,通过投票验证模型参数的正确性,可以抵抗后门数据集。Zhao 等[12]在训练过程中采用一种功能机制扰动神经网络的目标函数,实现差分隐私。由于在纵向联邦学习中,有

毒嵌入向量在稀疏性和多样性上与良性嵌入向量几乎没有差别,因此上述基于相似度或者聚类算法的抵御方案并不适用于 VFL。在 VFL 中,Qiu 等[13]通过向嵌入向量中添加扰动来建立扰动和目标标签之间的联系,进而干扰模型推理结果。He 等[14]用尽可能远离非目标类的触发向量替换目标样本的嵌入向量,从而误导顶部模型产生错误的结果。目前针对纵向联邦学习非目标数据投毒攻击以及抵御的研究还未出现。Wang 等[15]提出使用效用评估的方式来拒绝使用质量低下的数据参与训练的被动方,但是直接丢弃效用评估结果较差的嵌入向量通常会导致模型训练更加偏向头部数据。本文设计了隐私保护的效用评估和权重计算协议,抵御被动方的数据投毒攻击,实现模型在恶意被动方场景特别是在长尾数据场景下的高收敛率和高准确率。

1.2　嵌入向量的隐私保护

在 SVFL 中,半诚实的主动方可以使用来自被动方的嵌入向量来推断其原始数据。在早期的研究中,有学者提出了隐私推理攻击[16],并采用同态加密[17]和差分隐私[18]来抵御推理攻击。虽然基于同态加密的方法可以为嵌入向量[19-22]提供有效的隐私保护,但昂贵的计算开销阻碍了这些方案应用于现实场景。为了提高效率,基于差分隐私的方法通过向嵌入向量[23-25]注入噪声来保护隐私。然而,上述基于差分隐私的方案会导致梯度计算不准确,从而导致性能损失。考虑到模型精度和计算成本的限制,Shi 等[26]使用了一种可去除的掩蔽机制来实现嵌入向量的安全聚合。Li 等[27]提出了一种基于拉格朗日编码计算和秘密共享的隐私保护前向聚合算法。Wang 等[28]通过将可去除的掩蔽机制引入本地嵌入过程来保护被动方的隐私。以上工作均基于半诚实威胁模型,忽略了参与方不可信的问题。本文基于掩蔽机制和 SHE 算法设计了隐私保护的损失计算协议,实现在恶意被动方场景下的嵌入向量安全聚合。

2　问题描述

2.1　系统模型

一个开放的纵向联邦学习环境由各种机构组成,如医疗保健机构、金融机构、政府机构等。在基于模型拆分的纵向联邦学习中,参与方被分类为主动方,被动方和服务器。如图 1 所示,主动方持有数据集的部分标签和顶部模型,可以通过部分标签计算训练损失。被动方负责提供数据特征,且不持有数据样本对应的实际标签,只拥有底层模型,负责将原始数据样本通过底层模型映射为嵌入向量。服务器可以是提供医疗、汽车或金融服务的机构,这些机构根据其需求和资源自主发布联邦学习任务,并生成公共参数。本文假设 VFL 中有 m 个主动方、n 个被动方和 1 个服务器。主动方和被动方分别用$\{A_1,A_2,\cdots,A_m\}$和$\{P_1,P_2,\cdots,P_n\}$表示。

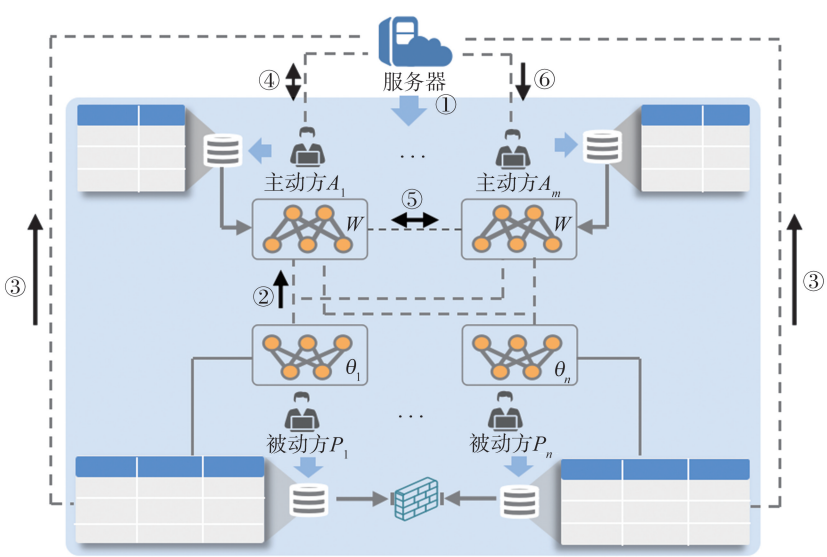

图 1　系统模型

Fig.1　System model

方案的工作流程如下:(1) 服务器发布一个纵向联邦学习任务,初始化顶层模型参数和被动方集合,并

将顶层模型发送给所有主动方；(2) 被动方生成随机值并将其发送给主动方；(3) 被动方掩蔽嵌入向量并将其发送给服务器；(4) 服务器和主动方协同计算损失，并评估每个被动方所发送嵌入向量的效用；(5) 服务器和主动方根据效用评估结果和被动方本地特征数量计算聚合权重；(6) 服务器和主动方对嵌入向量进行加权聚合，继续进行正向传播并更新顶层模型。

2.2 威胁模型

假设被动方是不可信的，为了阻碍模型收敛，被动方可能会发动数据投毒攻击，例如使用随机数替代训练样本生成嵌入向量，并发送给主动方和服务器以干扰顶层模型的训练。服务器和所有主动方是诚实且好奇的，他们诚实地执行纵向联邦学习训练过程，但是对被动方的私有数据感兴趣。具体来说，主动方与服务器接收到来自被动方的嵌入向量后，会根据协议正确执行效用评估和权重计算等操作，但是他们可能会对嵌入向量发动推理攻击来重建被动方的原始特征数据。

2.3 设计目标

(1) 抵抗数据投毒攻击。本文方案能够通过对嵌入向量的效用评估来检测和抵御恶意被动方发动的数据投毒攻击，并在聚合时丢弃有毒的嵌入向量。

(2) 抵抗隐私推理攻击。本文方案保护被动方的嵌入向量不受半诚实主动方和服务器的推理攻击，即它们无法通过推理被动方所发送的消息来获得任何关于特征数据的信息。

(3) 长尾数据下高收敛率。本文方案不仅适用于样本类别均匀分布的普通场景，在长尾数据场景下也具有较高的模型收敛率和准确率，主动方和服务器能正确区分有毒样本和尾部样本所对应的嵌入向量并赋予相应的聚合权重。

3 算法设计

3.1 技术概述

本文方案主要包括 3 个阶段，分别是系统初始化阶段、隐私保护的损失计算阶段和效用评估与权重计算阶段，方案流程如图 2 所示。为了降低多主动方和服务器协同进行模型推理的开销，方案在系统初始化阶段提出一种预计算的线性层计算方式。在文献[29]的基础上将算法由两方扩展到多方，让被动方选择随机数发送给多个主动方。主动方预先计算每层模型参数与随机数的内积，使得训练开始后线性层的推理与在明文上计算的开销相同，且被动方只需要与服务器进行交互，避免了训练过程中被动方与主动方之间频繁的交互过程。

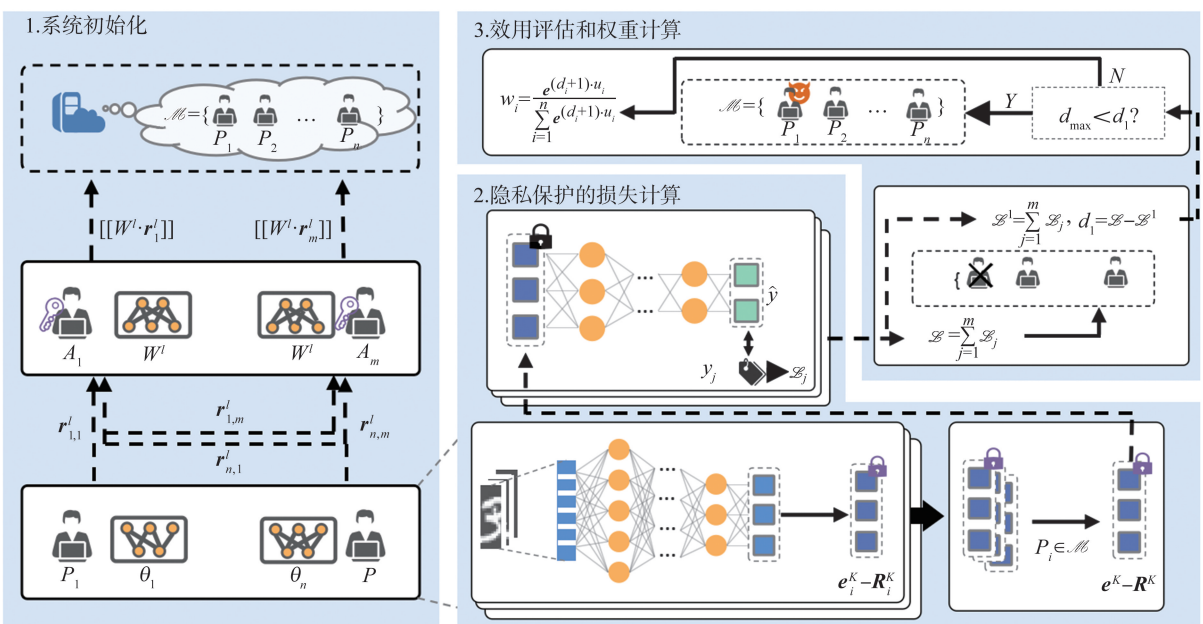

图 2 方案流程

Fig.2 Work flow

在隐私保护的损失计算阶段,为了降低多参与方非线性层的隐私推理开销,首先将顶层模型的非线性激活函数近似为二阶多项式。由于传统的基于海狸三元组的多项式评估算法在面向多个参与方时计算开销会呈现指数增长,所以本文设计一种基于对称同态加密的隐私保护损失计算协议。服务器对添加掩蔽后的嵌入向量进行加密后,基于 SHE 的同态加法和乘法属性,直接进行密态多项式评估。将评估结果发送给主动方进行解密,由于解密结果是掩蔽后的嵌入向量,因此所有参与方都无法得评估结果。

在效用评估与权重计算阶段,为了在长尾数据场景下抵御恶意被动方发动的数据投毒攻击,首先通过计算最大容忍距离来作为衡量中毒数据和尾部数据的阈值,在保留尾部数据样本所对应的嵌入向量的前提下剔除有毒样本对应的嵌入向量。然后基于被动方所持有的数据特征量和效用评估结果设计自适应权重计算协议,避免顶层模型偏向头部样本,加强对尾部样本的学习。

3.2　方案细节

3.2.1　系统初始化阶段

此阶段主要完成服务器发布公共参数和被动方与主动方交互掩蔽向量的工作,主要过程如协议 1 所示。服务器初始化顶层模型参数 W,维护一个被动方集合 $\mathscr{M}$,将 W 发送给所有主动方。被动方 P_i 初始化底层模型参数 $\theta_i \in \mathbf{R}^{d_i \times u}$,生成用于掩蔽的随机向量 $\boldsymbol{r}_{i,j}^l \in \mathbf{R}^n$ 并发送 $\boldsymbol{r}_{i,j}^l$ 给主动方 A_j。为了避免在训练阶段被动方与主动方之间频繁的交互,A_j 预先计算第 l 个线性层输出的份额 $W^l \cdot \boldsymbol{r}_{i,j}^l$ 并聚合得到 $W^l \cdot \boldsymbol{r}_j^l$,其中 W^l 是顶层模型 W 的第 l 层模型参数。为了进行对称同态加密,所有主动方需要共同的密钥,A_j 的公私钥对为 (pk_j, sk_j),所有主动方通过多方 Diffie-Hellman 密钥协商协议,得到共享密钥 s,并选择最大的两个素数 p 和 q。A_j 通过伪随机数生成器(pseudorandom generator, PRG)生成随机数 λ,并且基于 SHE 加密 $W^l \cdot \boldsymbol{r}_j^l$ 得到线性层输出份额的密文 $[[W^l \cdot \boldsymbol{r}_j^l]]$。$A_j$ 发送密文 $[[W^l \cdot \boldsymbol{r}_j^l]]$ 到服务器,服务器聚合得到密文 $[[W^l \cdot \boldsymbol{R}^l]]$。

3.2.2　隐私保护的损失计算阶段

此阶段主要实现多个主动方与服务器在持有嵌入向量份额的前提下协同完成顶层模型的推理过程,主要过程如协议 1 所示。

被动方 P_i 将数据样本 $x_i \in \boldsymbol{R}^{d_i}$ 输入到底层模型 θ_i,经过正向传播过程得到嵌入向量 $\boldsymbol{e}_i^K = \mathscr{F}_i(x_i, \theta_i)$,其中底层模型的第 K 层为剪切层。为了保护嵌入向量的隐私,P_i 用随机向量 $\boldsymbol{R}_i^K$ 对 $\boldsymbol{e}_i^K$ 进行掩蔽得到掩蔽后的嵌入向量 $\boldsymbol{e}_i^K - \boldsymbol{R}_i^K$,其中并将掩蔽后的嵌入向量发送给服务器。此时主动方持有份额 $\boldsymbol{r}_{i,j}^K$,服务器持有份额 $\boldsymbol{e}_i^K - \boldsymbol{R}_i^K$,服务器和主动方在本地将各自持有的份额输入到底层模型,协同进行模型推理计算损失。为了协同计算第 K 个线性层,服务器计算线性层输出的份额 $W^K \cdot (\boldsymbol{e}_i^K - \boldsymbol{R}_i^K)$,并聚合得到 $W^K \cdot (\boldsymbol{e}^K - \boldsymbol{R}^K)$。此时,每个主动方和服务器分别持有第 K 个线性层输出份额 $W^K \cdot \boldsymbol{r}_j^K$ 和 $W^K \cdot (\boldsymbol{e}^K - \boldsymbol{R}^K)$,但此时他们不需要交互自己持有的结果,而是直接将结果作为非线性层的输入份额。

协议 1　隐私保护的损失计算协议

输入: W^l(第 l 层顶层模型); $\boldsymbol{e}_i^K$(P_i 的嵌入向量); $\mathscr{M}$(被动方集合); $\{\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{m}_2\}$(多项式拟合系数)

① P_i 生成并发送 $\boldsymbol{r}_{i,j}^l$ 到主动方 A_j。

② A_j 计算 $W^l \cdot \boldsymbol{r}_{i,j}^l$ 和 $W^l \cdot \boldsymbol{r}_j^l = \sum_{i=1}^{n} W^l \cdot \boldsymbol{r}_{i,j}^l$。

③ 所有主动方计算 $S = g^{\prod_{j=1}^{m} sk_j}$,并选择 $p, q \in [S]$。

④ A_j 计算 $\lambda = \mathrm{PRG}(S)$ 和 $[[W^l \cdot \boldsymbol{r}_j^l]] = \mathrm{SHE.Enc}(W^l \cdot \boldsymbol{r}_j^l, \mathrm{key})$,其中 $\mathrm{key} = (p, \lambda)$。

⑤ A_j 发送到服务器。

⑥ 服务器计算 $[[W^l \cdot \boldsymbol{R}^l]] = \sum_{j=1}^{m} [[W^l \cdot \boldsymbol{r}_j^l]]$。

⑦ P_i 生成 $\boldsymbol{e}_i^K - \boldsymbol{R}_i^K$,其中 $\boldsymbol{R}_i^K = \sum_{j=1}^{m} \boldsymbol{r}_{i,j}^K$,将 $\boldsymbol{e}_i^K - \boldsymbol{R}_i^K$ 其发送给服务器。

⑧ for 顶层模型的第 $l \in K, K+1, \cdots, L$ 层 do

⑨　　服务器计算 $W^l \cdot (\boldsymbol{e}^l - \boldsymbol{R}^l) = \sum_{i=1}^{n} W^l \cdot (\boldsymbol{e}_i^l - \boldsymbol{R}_i^l)$,并将其发送给主动方。

⑩ A_j 计算$[[W^l\cdot(\boldsymbol{e}^l-\boldsymbol{R}^l)]]=\mathrm{SHE.Enc}(W^l\cdot(\boldsymbol{e}^l-\boldsymbol{R}^l),\mathrm{key})$,并将其发送给服务器。

⑪ 服务器计算$[[W^l\cdot\boldsymbol{e}^l]]=[[W^l\cdot(\boldsymbol{e}^l-\boldsymbol{R}^l)]]+[[W^l\cdot\boldsymbol{R}^l]]$,评估多项式$[[\boldsymbol{e}^{l+1}]]=[[m_0]]+m_1\cdot[[W^l\cdot\boldsymbol{e}^l]]+m_2\cdot[[W^l\cdot\boldsymbol{e}^l]]^2$。

⑫ 服务器计算$[[\boldsymbol{e}^{l+1}-\boldsymbol{R}^{l+1}]]$,将其发给所有主动方。

⑬ A_j 计算 $\boldsymbol{e}^{l+1}-\boldsymbol{R}^{l+1}=\mathrm{SHE.Dec}([[\boldsymbol{e}^{l+1}-\boldsymbol{R}^{l+1}]],\mathrm{key})$,并将其发送给服务器。

⑭ end

⑮ 服务器计算 $\hat{\boldsymbol{y}}=\boldsymbol{e}^{L+1}$并将其发送给所有主动方。

⑯ A_j 计算 $\mathscr{L}_j=\frac{1}{2}(\boldsymbol{y}_j-\hat{\boldsymbol{y}})^2$, $j\in\{1,m\}$并将 $\mathscr{L}_j$ 发送到服务器。

⑰ 服务器计算 $\mathscr{L}=\sum_{j=1}^{m}\mathscr{L}_j$。

为了协同计算第 K 个非线性层,主动方与服务器将 ReLU 函数近似为一个二阶多项式 $f(x)=m_0+m_1x+m_2x^2$。如图 3 所示,服务器发送线性层输出份额 $W^K\cdot(\boldsymbol{e}^K-\boldsymbol{R}^K)$给所有主动方,主动方 A_j 基于 SHE 加密得到线性层输出份额的密文$[[W^K\cdot(\boldsymbol{e}^K-\boldsymbol{R}^K)]]$,并将其返回给服务器。服务器基于 SHE 的同态加法性质计算线性层输出的密文$[[W^K\cdot\boldsymbol{e}^K]]$,并评估多项式 $f(x)$ 得到非线性层输出的密文$[[\boldsymbol{e}^{K+1}]]$。由于服务器不持有密钥,因此无法解密该密文。于是对密文添加掩蔽后得到$[[\boldsymbol{e}^{K+1}-\boldsymbol{R}^{K+1}]]$并将其发送给所有主动方。主动方 A_j 解密得到 $\boldsymbol{e}^{K+1}-\boldsymbol{R}^{K+1}$,并将其返回给服务器。此时,服务器与主动方们又分别持有了第 $K+1$个线性层的输入份额 $\boldsymbol{e}^{K+1}-\boldsymbol{R}^{K+1}$和 $\boldsymbol{r}_j^{K+1}$。服务器与所有主动方可以协同执行第 $K+1$ 层的线性层运算。直到第 L 层计算结束后,服务器获得最终输出的份额 $\boldsymbol{e}^{L+1}-\boldsymbol{R}^{L+1}$并接收来自所有主动方发送的份额 $\boldsymbol{r}_j^{L+1}$, $j\in\{1,2,\cdots,m\}$,恢复输出结果 $\boldsymbol{e}^{L+1}=\boldsymbol{e}^{L+1}-\boldsymbol{R}^{L+1}+\sum_{j=1}^{m}\boldsymbol{r}_j^{L+1}$,其中预测结果为 $\hat{\boldsymbol{y}}=\boldsymbol{e}^{L+1}$,将预测结果 $\hat{\boldsymbol{y}}$ 发送给所有主动方。主动方 A_j 根据 $\hat{\boldsymbol{y}}$ 和本地持有的标签 $\boldsymbol{y}_j$ 计算本地训练损失 $\mathscr{L}_j$,并将 $\mathscr{L}_j$ 发送给服务器。最后,服务器计算总损失值 $\mathscr{L}$。

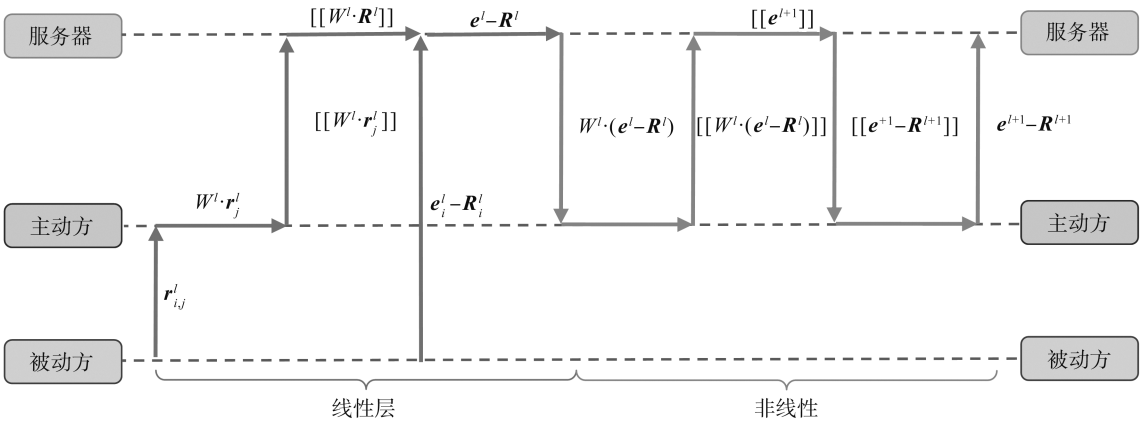

图 3 隐私保护的损失计算流程

Fig.3 Privacy protection loss calculation process

3.2.3 效用评估与权重计算阶段

协议 2 效用评估和权重计算协议

输入: W^l(第 l 层的顶层模型);$\boldsymbol{e}_i^K$(P_i 的嵌入向量);$\mathscr{M}$(被动方集合); D(距离集合);θ_i(P_i 的底层模型)

① 服务器通过协议 1 计算 $\mathscr{L}$。

② for 每个被动方 P_i, $i\in1,\cdots,n$ do

③ 服务器排除 P_i 并更新 $\mathscr{M}^i$,其中 $\mathscr{M}^i=\mathscr{M}\setminus\{P_i\}$。

④ 服务器通过协议 1 计算 $\mathscr{L}^i$。

⑤ 服务器计算 $d_i=\mathscr{L}-\mathscr{L}^i$。

⑥ if $d_i>0$ then

⑦　$D=D\cup\{d_i\}$

⑧　end

⑨ end

⑩ 服务器计算 $d_{\max}=\dfrac{\sum_{i=1}^{n}d_i}{n-\bar{n}}$,其中 $\bar{n}=|D|$。

⑪ if $d_i<0$ then

⑫　服务器计算 $w_i=\dfrac{e^{u_i/|d_i-1|}}{\sum_{i=1}^{n}e^{u_i/|d_i-1|}}$。

⑬ else if $0<d_i<d_{\max}$

⑭　服务器计算 $w_i=\dfrac{e^{(d_i+1)\cdot u_i}}{\sum_{i=1}^{n}e^{(d_i+1)\cdot u_i}}$。

⑮ else

⑯　服务器计算 $w_i=0$。

⑰ end

⑱　A_j 聚合 $W^l\cdot\boldsymbol{r}_j^l=W^l\cdot\sum_{i=1}^{n}w_i\cdot\boldsymbol{r}_{i,j}^l$,其中 $\{P_i\}\notin\mathscr{M}^*$。

⑲　服务器聚合 $W^l\cdot(\boldsymbol{e}^l-\boldsymbol{R}^l)=W^l\cdot\sum_{i=1}^{n}w_i\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)$,其中 $\{P_i\}\notin\mathscr{M}^*$。

此阶段主要实现主动方与服务器对被动方所发送的嵌入向量的效用评估,并根据评估结果进行聚合权重的计算,主要过程如协议 2 所示。

为了评估被动方 P_i 发送的嵌入向量 $\boldsymbol{e}_i^K$ 的效用,服务器将 P_i 从集合 $\mathscr{M}$ 中排除,即 $\mathscr{M}^i=\mathscr{M}\setminus\{P_i\}$。服务器和主动方计算 $W^l\cdot\boldsymbol{r}_j^l$ 和 $W^l\cdot(\boldsymbol{e}^l-\boldsymbol{R}^l)$,根据上述步骤共同计算出不包含嵌入向量 $\boldsymbol{e}_i^K$ 的训练损失 $\mathscr{L}^i$。服务器计算距离 d_i,并判断 d_i 是否大于 0。如果 $d_i>0$,则服务器将 d_i 加入集合 D,即 $D=D\cup\{d_i\}$。服务器计算最大容忍距离 $d_{\max}$,其中 $|D|=\bar{n}$,并根据 $d_{\max}$ 对嵌入向量 $\boldsymbol{e}_i^K$ 计算聚合权重 w_i。当 $d_i<0$ 时,服务器认为 $\boldsymbol{e}_i^K$ 是对模型收敛有正向贡献的头部样本所对应的嵌入向量,因此为其赋予较大权重

$$w_i=\frac{e^{u_i/|d_i-1|}}{\sum_{i=1}^{n}e^{u_i/|d_i-1|}},\tag{1}$$

其中 u_i 是 P_i 的训练数据集中的特征数量。当 $0<d_i<d_{\max}$ 时,服务器认为 $\boldsymbol{e}_i^K$ 是尾部样本所对应的嵌入向量,为了保证顶层模型能够充分学习到该类样本,服务器为其赋予权重

$$w_i=\frac{e^{(d_i+1)\cdot u_i}}{\sum_{i=1}^{n}e^{(d_i+1)\cdot u_i}}。\tag{2}$$

当 $0<d_i<d_{\max}$ 时,服务器认为 $\boldsymbol{e}_i^K$ 是有毒样本所对应的嵌入向量,为了保证顶层模型训练的准确性,服务器为其赋予权重 $w_i=0$。主动方 A_j 和服务器分别添加权重 w_i 得到加权后的本地份额 $W^l\cdot\boldsymbol{r}_j^l$ 和 $W^l\cdot(\boldsymbol{e}^l-\boldsymbol{R}^l)$。主动方 A_j 根据协议 1 协同计算损失 $\mathscr{L}_j$ 和顶层模型的梯度 ∇W_j,并将 ∇W_j 发送给服务器。服务器聚合得到梯度 ∇W 并发送给所有主动方。主动方 A_j 更新顶层模型 $W=W+\eta_j\nabla W$,其中 η_j 是 A_j 的学习率,然后将切层的梯度 S_j 发送给所有被动方。被动方 P_i 计算底层模型的梯度 $\nabla\theta_i=\mathscr{G}_i(S_j,\theta_i)$,然后更新底层模型参数 $\theta_i=\theta_i+\eta_i\nabla\theta_i$,其中 η_i 是 P_i 的学习率。

4　隐私性分析

定理　基于算术秘密分享和对称同态加密的隐私性,本文方案可以抵御主动方的数据推理攻击,保护被

动方的本地数据隐私。对于任意敌手 $\mathscr{C}_a$,存在模拟器 $\mathscr{C}_s$ 使得以下关系成立

$$\Pr\left[\mathscr{A}(1^\lambda,m)=\mathscr{O}(1^\lambda,\boldsymbol{e}_i^K)\ \middle|\ \begin{pmatrix}\boldsymbol{e}_i^K-\boldsymbol{R}_i^K\\ W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)\\ W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)\\ \boldsymbol{e}_i^{l+1}-\boldsymbol{R}_i^{l+1}\end{pmatrix}\leftarrow\text{协议 1}\right]$$

$$\approx\Pr\left[\mathscr{A}(1^\lambda,m^*)=\mathscr{O}(1^\lambda,\boldsymbol{e}_i^K)\ \middle|\ \begin{pmatrix}\boldsymbol{e}_i^*-\boldsymbol{R}_i^K\\ W^l\cdot(\boldsymbol{e}_i^*-\boldsymbol{R}_i^l)\\ W^l\cdot(\boldsymbol{e}_i^*-\boldsymbol{R}_i^l)\\ [[\boldsymbol{e}_i^*-\boldsymbol{R}_i^{l+1}]]\end{pmatrix}\leftarrow\mathrm{Sim}(W^l,\mathscr{M})\right],$$

其中,$\mathscr{O}:\{0,1\}^*\to\{0,1\}^*$ 是多项式有界函数,$\mathscr{O}(1^\lambda,\boldsymbol{e}_i^K)$ 的输出表示关于嵌入向量的任何隐私信息,$\mathscr{A}(1^\lambda,m)$ 表示 $\mathscr{C}_a$ 在安全参数 1^λ 上的输出,$m=(\boldsymbol{e}_i^K-\boldsymbol{R}_i^K,W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l),[[W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)]],[[\boldsymbol{e}_i^{l+1}-\boldsymbol{R}_i^{l+1}]])$。

证明 假设最多 $m-1$ 个主动方、$n-\kappa$ 个被动方和服务器被敌手 $\mathscr{C}_a$ 控制。$\mathscr{C}_s$ 运行如下:

(1) $\mathscr{C}_s$ 为 $\mathscr{C}_a$ 选择一个均匀分布的随机状态,

(2) $\mathscr{C}_s$ 通过选择长度为 u 的正态分布随机数生成 P_i 的虚拟嵌入向量 $\boldsymbol{e}_i^*$, $i\in\{1,\cdots,\kappa\}$,

(3) $\mathscr{C}_s$ 生成随机值 $\boldsymbol{R}_i^K$,并将 $\boldsymbol{e}_i^*-\boldsymbol{R}_i^K$ 发送给 $\mathscr{C}_a$,

(4) $\mathscr{C}_s$ 计算第 l 层线性层的输出 $W^l\cdot(\boldsymbol{e}_i^*-\boldsymbol{R}_i^l)$,其中 $l\in[K,N]$,并将其发送给 $\mathscr{C}_a$,

(5) $\mathscr{C}_s$ 通过 SHE 计算密文$[[W^l\cdot(\boldsymbol{e}_i^*-\boldsymbol{R}_i^l)]]$,并将其发送给 $\mathscr{C}_a$,

(6) $\mathscr{C}_s$ 通过评估近似多项式 $f(x)$ 计算第 l 层非线性层的输出$[[\boldsymbol{e}_i^*-\boldsymbol{R}_i^{l+1}]]$,并将其发送给 $\mathscr{C}_a$,

(7) $\mathscr{C}_s$ 通过 SHE 计算明文 $\boldsymbol{e}_i^*-\boldsymbol{R}_i^{l+1}$,并将其发送给 $\mathscr{C}_a$。

本文利用一个混合论证来证明 $\mathrm{Sim}_{\mathscr{A}}$ 的输出在计算上与 $\mathrm{Real}_{\mathscr{A}}$(敌手 $\mathscr{C}_a$ 在真实世界中的视角)的输出不可区分,即 $\mathrm{Real}_{\mathscr{A}}\overset{c}{\equiv}\mathrm{Sim}_{\mathscr{A}}$。

H_0:对应于真实世界的分布。

H_1:与 H_0 相同,除了在(2)中将 $\boldsymbol{e}_i^K$ 替换为虚拟的 $\boldsymbol{e}_i^*$,并在(3)中用 $\boldsymbol{R}_i^K$ 进行掩码。由于基于算术秘密共享的安全性,掩码后的嵌入向量的分布在计算上与对手从 $\mathrm{Real}_{\mathscr{A}}$ 中观察到的不可区分。这保证了 $\boldsymbol{e}_i^*-\boldsymbol{R}_i^K$ 和 $\boldsymbol{e}_i^K-\boldsymbol{R}_i^K$ 具有相同的分布。因此,H_0 和 H_1 在计算上不可区分。

H_2:与 H_1 相同,除了在(4)中将第 l 层线性层的输出 $W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)$ 替换为 $W^l\cdot(\boldsymbol{e}_i^*-\boldsymbol{R}_i^l)$。因为作为对手的主动方 $A_{j\in[1,m-1]}$ 仅持有单个 $\boldsymbol{r}_j^K$,掩码 $\boldsymbol{R}_i^K$ 无法被移除。这保证了 $W^l\cdot(\boldsymbol{e}_i^*-\boldsymbol{R}_i^l)$ 和 $W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)$ 具有相同的分布。因此,H_1 和 H_2 在计算上不可区分。

H_3:与 H_2 相同,除了在(5)中将密文$[[W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)]]$替换为$[[W^l\cdot(\boldsymbol{e}_i^*-\boldsymbol{R}_i^l)]]$。因为作为对手的服务器不持有密钥,掩码$[[W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)]]$无法被解密。这保证了$[[W^l\cdot(\boldsymbol{e}_i^l-\boldsymbol{R}_i^l)]]$和$[[W^l\cdot(\boldsymbol{e}_i^*-\boldsymbol{R}_i^l)]]$具有相同的分布。因此,$H_2$ 和 H_3 在计算上不可区分。

H_4:与 H_3 相同,除了在(6)中将$[[\boldsymbol{e}_i^{l+1}-\boldsymbol{R}_i^{l+1}]]$替换为$[[\boldsymbol{e}_i^*-\boldsymbol{R}_i^{l+1}]]$,并在(7)中将 $\boldsymbol{e}_i^{l+1}-\boldsymbol{R}_i^{l+1}$ 替换为 $\boldsymbol{e}_i^*-\boldsymbol{R}_i^{l+1}$。如 H_2 所述,$[[\boldsymbol{e}_i^{l+1}-\boldsymbol{R}_i^{l+1}]]$和$[[\boldsymbol{e}_i^*-\boldsymbol{R}_i^{l+1}]]$具有相同的分布。如 H_1 所述,$\boldsymbol{e}_i^{l+1}-\boldsymbol{R}_i^{l+1}$ 和 $\boldsymbol{e}_i^*-\boldsymbol{R}_i^{l+1}$ 也具有相同的分布。因此,H_3 和 H_4 在计算上不可区分。

最后,可以得到 H_3 与 $\mathrm{Sim}_{\mathscr{A}}$ 的输出是同分布的,即 $\mathrm{Sim}_{\mathscr{A}}$ 的输出在计算上与 $\mathrm{Real}_{\mathscr{A}}$ 的输出不可区分,证明完成。

5 实验

5.1 实验设置

本文使用 Python 和 PyTorch 库进行单机模拟和实验仿真,运行云服务器配备 RTX 4060 GPU 和

62 GB内存,基于 NIST P-256 曲线的椭圆曲线 Diffie-Hellman 算法实现密钥协商。本文使用 MNIST、Fashion-MNIST、CIFAR-10 和 Yahoo!Answers 数据集。MNIST 和 Fashion-MNIST 数据集都包含 70 000 张 28 * 28 像素的灰色图像,其中 60 000 个图像是训练样本,10 000 个图像是测试样本。CIFAR-10 数据集包含 10 个类别的 32 * 32 像素的彩色图像,每个类别分别有 5 000 个训练样本和 1 000 个测试样本。“Yahoo!Answers”分类数据集包含 10 个主要类别,每个类别有 140 000 个训练样本和 6 000 个测试样本。

本方案使用学习率为 0.01 的小批量梯度下降优化器。训练轮次固定为 100 轮,每个被动方的本地批次大小设置为 64。在 MNIST、CIFAR-10 以及 Fashion-MNIST 数据集下,训练一个包含两个卷积层和三个全连接层的卷积神经网络。其中,该卷积神经网络被拆分为底层模型和顶层模型,底层模型包含两个卷积层和一个全连接层,在没有特殊说明的情况下,顶层模型包含两个全连接层。在“Yahoo!Answers”数据集下,训练一个包含三个全连接层的 MLP 网络。其中底层模型包含两个全连接层,在没有特殊说明的情况下,顶层模型包含一个全连接层。为了模拟数据投毒攻击,设置一个恶意被动方,其数据投毒比例为 0.5。例如,恶意被动方将其所持有的 60 000 个训练样本中的 30 000 个训练样本特征用随机值代替。为了模拟现实的数据分布,本文在多个参与方之间分配特征和标签,采用非独立同分布的方式。对于 MNIST 数据集特征非独立同分布,每个样本被分成 3 部分:1 * 28 * 4、1 * 28 * 10 和 1 * 28 * 14,分别分配给被动方 P_1、P_2 和 P_3。对于标签非独立同分布,主动方 A_1 持有类别{0,1,2,3,4}的标签,而 A_2 持有类别{5,6,7,8,9}的标签。此外,为了模拟长尾数据场景,需要确保少数类占据数据集样本的多数。具体来说,对于 MNIST 数据集,保留类别{0,4,8}的所有样本,而类别{1,2,3,5,6,7,9}的样本仅保留 1/10。对于其他数据集,本文执行相同的操作以模拟类似条件。

为了验证本文方案能够有效抵御数据投毒攻击,并且在长尾数据场景下有较高的模型收敛率和准确率,本文与以下 3 个基线方案进行了对比。

(1) BDVFL[20]:该方案通过对嵌入向量进行效用评估来衡量被动方本地数据质量,通过直接丢弃质量评估结果较差的嵌入向量来提高模型的收敛率和准确率。

(2) FedVS[19]:该方案基于拉格朗日编码计算来实现被动方掉线鲁棒和嵌入向量的隐私保护,嵌入向量以码字形式被平均聚合后解码输入到顶层模型。

(3) CVFL[30]:该方案首次提出多主动方的纵向联邦学习架构,并基于优化算法解决参与方掉队问题。

5.2　计算开销

评估不同主动方数量和顶层模型规模下的系统初始化阶段的计算开销。

如图 4(a)所示,当主动方数量为 2、4、6 时,密钥协商需要的时间分别为 2.05、8.33、20.94 ms,同态加密的时间分别仅为 0.57、0.66、0.59 ms。由于密钥协商需要主动方相互之间交换公钥,所以会随着主动方数量的增长而增长。此外,固定主动方数量 $m=3$,随着顶层模型的层数为 1、2、3、4 时,密钥协商开销不变,同态加密开销分别为 0.71、1.53、7.49、47.51 ms。这是因为,当顶层模型尺寸增加时,需要进行对称同态加密的向量规模也会大幅增加。

在损失计算阶段,主动方和服务器协同进行顶层模型的正向传播过程,该阶段的计算开销与嵌入向量的长度和顶层模型的尺寸呈线性相关。在顶层模型只有 1 层且嵌入向量长度为 500、1 500、2 500 时,主动方侧损失计算开销分别为 16.77、64.56、134.59 ms。当嵌入向量长度为 1 000 且顶层模型层数为 1、2、3、4 时,主动方侧损失计算开销分别为 29.76、34.25、41.85 和 46.13 ms。因为主动方侧同态加密操作占据开销的主要部分,且更高的嵌入向量长度和顶层模型尺寸都会使需要加密的明文规模增加。

此外,服务器侧的损失计算开销同样与嵌入向量的长度和顶层模型的尺寸呈线性相关。如图 5 所示,本文分别评估了线性层和非线性层的计算开销,固定顶层模型层数 $l=1$,当嵌入向量长度为 500、1 500、2 500 时,线性层的计算开销分别为 0.041、0.112、0.196 ms,非线性层分别为 5.76、17.25、28.64 ms,比线性层高两个数量级。这是因为相比于线性层接近明文的运算速度,非线性层包括大量的 SHE 的同态加法和乘法。

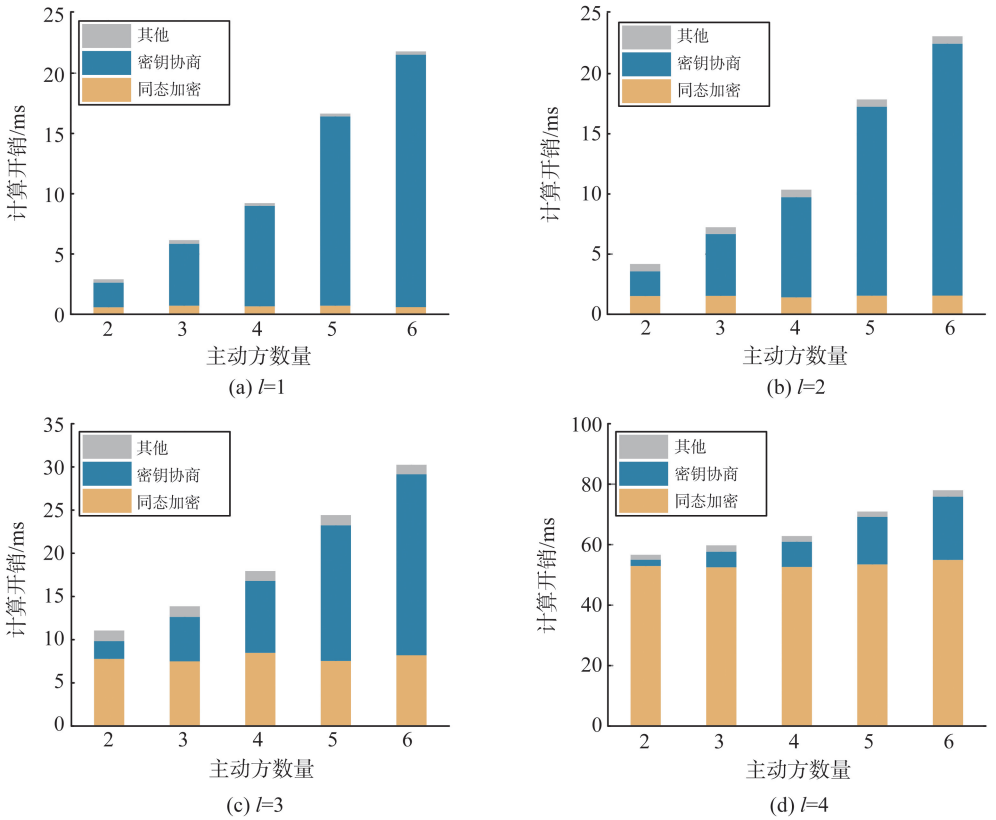

图 4 不同尺寸顶层模型下系统初始化阶段的计算开销

Fig.4 Computational cost of the system initialization phase for different sized top models

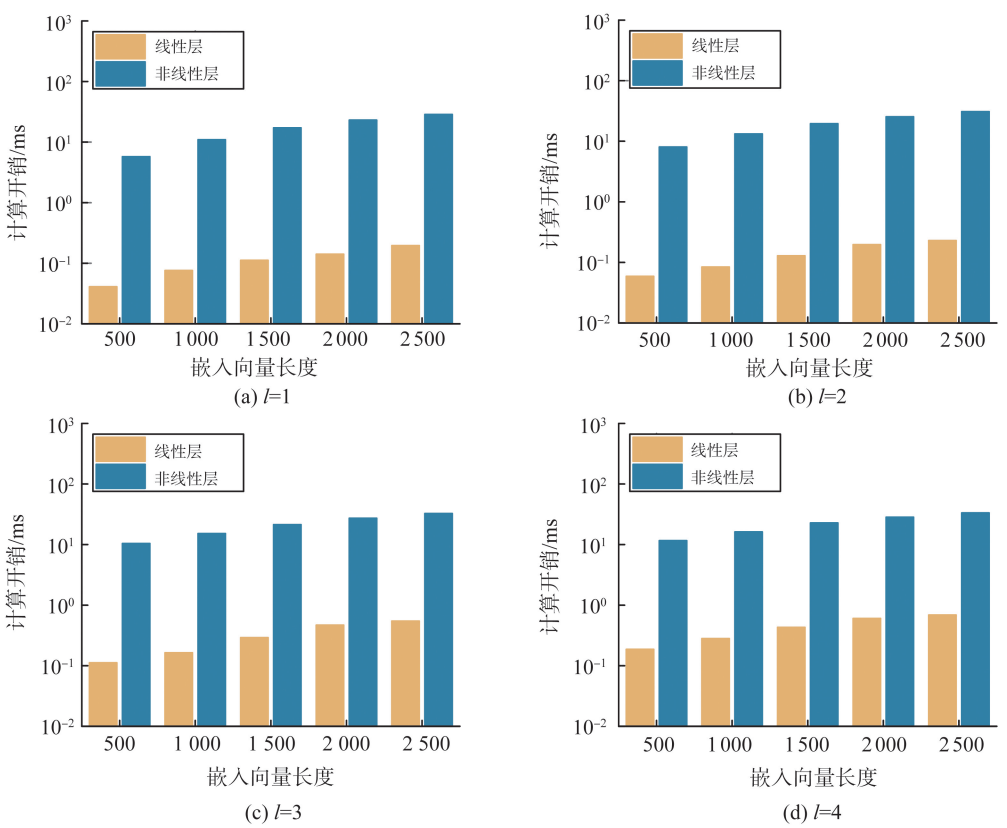

图 5 损失计算阶段服务器侧的计算开销

Fig.5 Computational cost on the server side of the loss calculation phase

效用评估的计算开销与被动方数量和嵌入向量长度呈正相关。如图 6(a)所示,当嵌入向量长度为 500、1 500、2 500 时,效用评估的计算开销分别为 45.06、163.61、326.46 ms。这是因为嵌入向量长度越高,协

同损失计算的开销就越大,进而导致更高的效用评估开销。固定嵌入向量长度为 1 000,当被动方数量为 2、3、4、5 时,效用评估开销为 81.44、121.63、162.41、203.96 ms。由于每一次只对单个被动方进行评估,所以被动方的数量直接影响效用评估的轮数。此外,权重计算的开销仅与被动方数量有关,当被动方数量为 2、3、4、5 时,权重计算开销仅需约 0.2、0.3、0.45、0.6 ms。

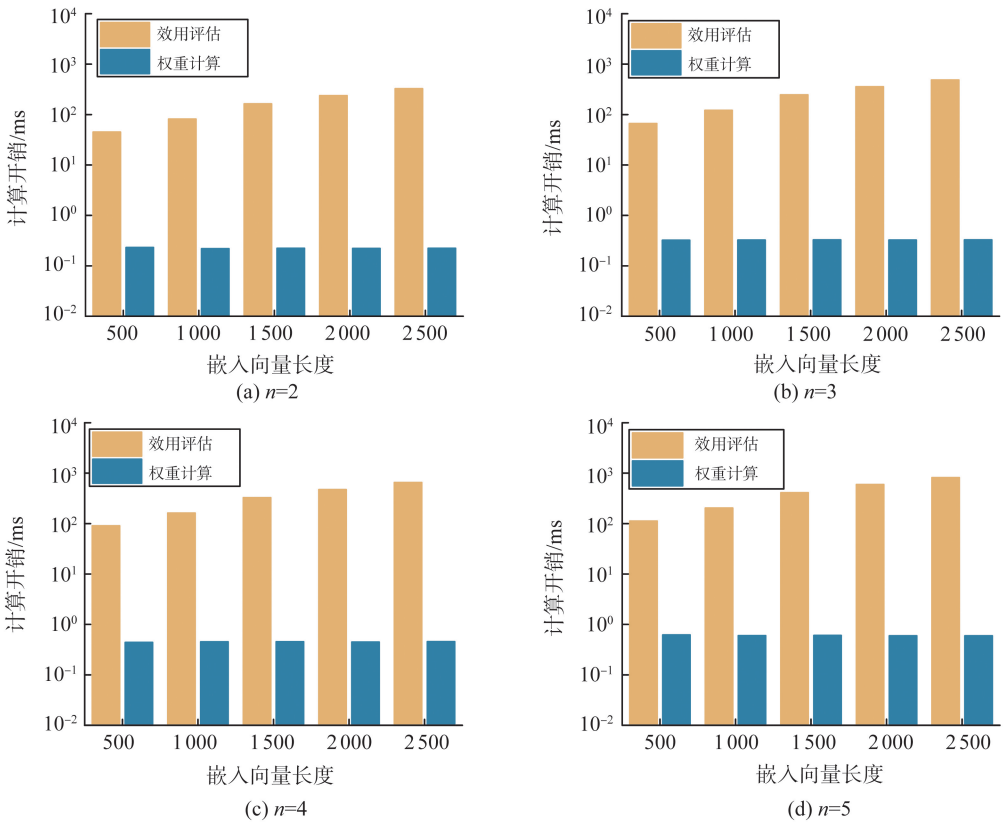

图 6　效用评估与权重计算阶段的计算开销

Fig.6　Computational cost of the utility evaluation and weight calculation phase

5.3　通信开销

本节评估了系统初始化阶段和隐私保护的损失计算阶段的通信开销。参与方之间的通信开销大小主要与顶层模型尺寸有关,当顶层模型层数越多时,参与方之间的通信开销越大。如表 1 所示,当顶层模型层数 l 分别为 1、2、3 和 4 时,系统初始化阶段的通信开销分别为 1.39 、5.28 、7.47 和 9.96,这是由于随着顶层模型层数的增加,在系统初始化阶段被动方需要向主动方发送更多的随机数。此外,当顶层模型层数 l 分别为 1、2、3 和 4 时,隐私保护的损失计算阶段的通信开销分别为 2.93 、10.86 、16.05 、21.24 kB。由于主动方和服务器基于 SHE 协同计算损失,传输密文占大部分通信开销,并且随着顶层模型层数的增加,主动方与服务器之间要传输更多密文。

表 1　通信开销

Table 1　Communication overhead

顶层模型层数 l	系统初始化阶段开销/kB	隐私保护的损失计算阶段开销/kB
1	1.39	2.93
2	5.28	10.86
3	7.47	16.05
4	9.96	21.24

5.4　存储开销

本节评估了各阶段不同实体的存储开销。如表 2 所示,在系统初始化阶段,被动方不需要存储任何的消息,因此不存在存储开销。主动方的则需要存储预先计算的线性层份额,需要产生约 16.94 kB 的存储开销。服务器需要计算并存储线性层份额的密文,因此需要产生约 60.48 kB 存储开销。在隐私保护的损失计算阶段,被动方需要暂时存储掩蔽后的嵌入向量等待获得一个批次的嵌入向量后统一发送给服务器,在这过程中

只产生约 0.97 kB 存储开销。主动方需要存储线性层份额的密文,以供服务器执行密态下的多项式评估,需要产生约 39.75 kB 存储开销。服务器需要存储非线性层的输出密文,产生约 209 kB 存储开销。在效用评估和权重计算阶段,被动方不参与该阶段的任务,不消耗内存。主动方负责为接收到的份额添加权重并暂存在本地,产生 1.62 kB 存储开销。服务器负责计算和存储每个嵌入向量对应的权重,产生约 166.6 kB 的存储开销。

表 2 存储开销

Table 2 Storage overhead 单位:开销/kB

阶段	被动方	主动方	服务器
系统初始化阶段	0	16.94	60.48
隐私保护的损失计算阶段	0.97	39.75	209.10
效用评估和权重计算阶段	0	1.62	166.60

5.5 收敛率与准确率

本节首先在普通场景下评估了方案的测试损失。该场景中每个被动方所持有的每个类别训练样本的数量是均衡的。将本文方案与 FedVS 和 BDVFL 进行对比,结果如表 3 和图 7 所示,以 MNIST 数据集为例,本方案分别比 FedVS 提前约 5 轮达到 0.5 的损失值。并且最终收敛时的损失比 FedVS 低约 0.4。这是由于在恶意被动方场景中,FedVS 没有引入任何的嵌入向量检测机制,将有毒数据生成的嵌入向量直接聚合并输入到顶层模型,严重影响了模型的收敛率和准确率。而本文方案与 BDVFL 由于都基于效用评估算法设计了嵌入向量过滤协议,在数据投毒场景中都有较好的模型准确率表现。此外,以 CIFAR-10 数据集为例,本文方案比 BDVFL 提前 2 轮达到 1.4 的测试损失。这是因为 BDVFL 并未考虑基于本地特征数量的嵌入向量加权机制,将所有通过检测的嵌入向量都设置了统一的聚合权重,忽略了在本地特征数量不相等情况下,嵌入向量对顶层模型贡献的不平衡问题。

表 3 普通场景下抵御数据投毒攻击的测试准确率

Table 3 Test accuracy against data poisoning attacks in normal scenarios 单位:%

方案	MNIST	CIFAR-10	Fashion-MNIST	Yahoo! Answers
FedVS	87.96±1.30	61.40±2.45	81.40±1.93	89.03±1.03
BDVFL	91.12±1.63	65.63±2.61	84.63±1.34	92.32±1.12
CVFL	91.22±1.42	63.13±2.12	86.13±1.18	92.84±1.27
本方案	93.80±0.92	69.91±2.06	87.91±1.12	95.01±1.21

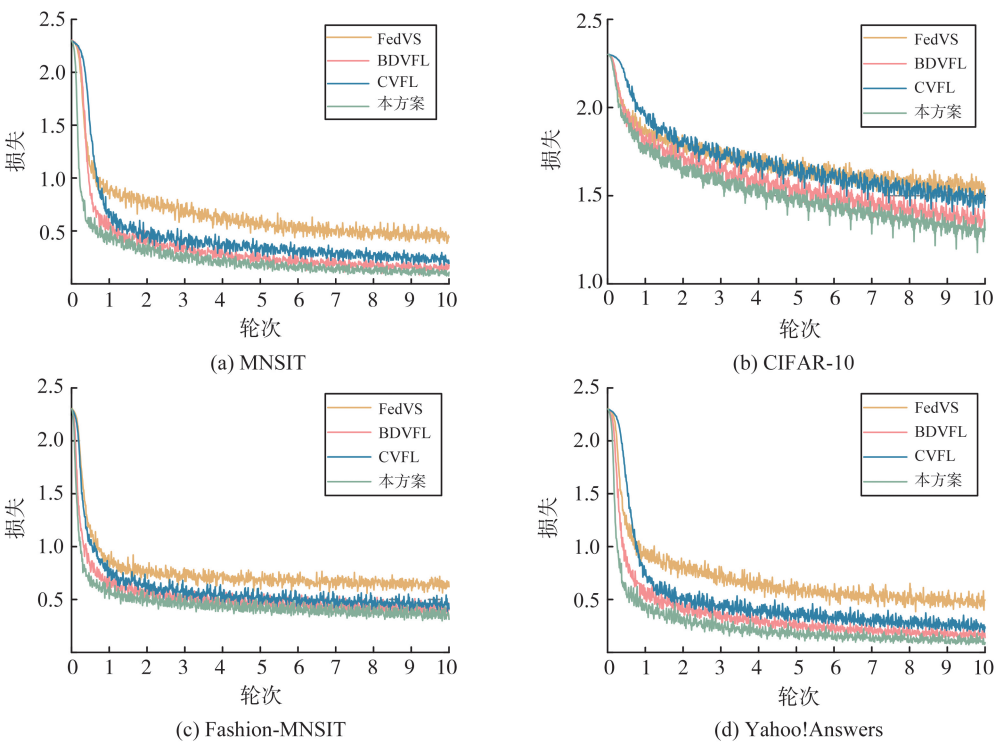

图 7 普通场景下抵御数据投毒攻击的测试损失

Fig.7 Test loss against data poisoning attacks in normal scenario

在长尾数据场景下,分别基于 4 个数据集评估本文方案的测试损失,并且与 FedVS 和 BDVFL 进行对比。评估结果如表 4 和图 8 所示,以 MNIST 数据集为例,本文方案分别比 FedVS 和 BDVFL 提前约 3 轮和 6 轮达到 0.75 的损失值。并且最终收敛时的损失比 FedVS 和 BDVFL 低约 0.2 和 0.3。这是由于在长尾数据以及恶意被动方场景中,FedVS 不加权直接聚合的操作不仅使得顶层模型未能充分学习到长尾数据,而且还将有毒的嵌入向量聚合并输入到顶层模型,阻碍模型的收敛率和准确率。而 BDVFL 会使主动方错误的将尾部样本所对应的嵌入向量归为有毒数据而丢弃,加剧顶层模型偏向数量更多的头部样本。

表 4　长尾数据场景下抵御数据投毒攻击的测试准确率

Table 4　Test accuracy against data poisoning attacks in long-tail data scenarios　　单位:%

方案	MNIST	CIFAR-10	Fashion-MNIST	Yahoo! Answers
FedVS	85.03±2.75	59.65±3.18	77.12±1.94	87.32±1.54
BDVFL	82.46±2.81	58.15±3.61	75.43±1.16	85.47±1.48
CVFL	85.94±2.41	60.69±2.96	79.76±1.96	89.04±1.89
本方案	89.07±2.03	65.46±2.43	85.16±1.31	92.01±1.98

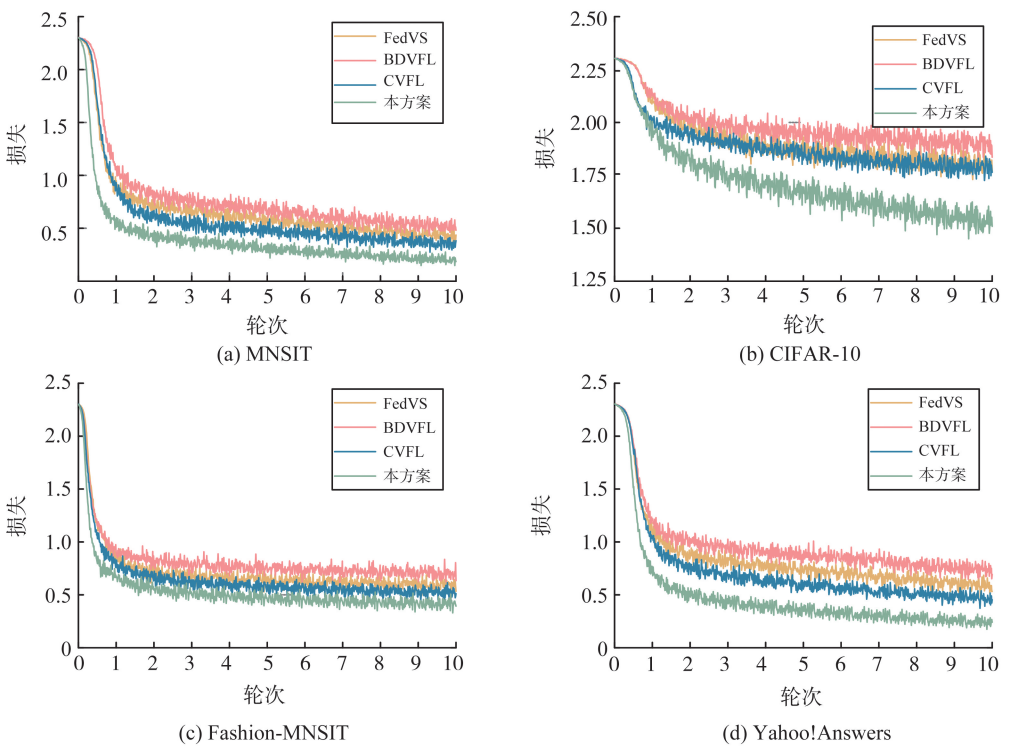

图 8　长尾数据场景下抵御数据投毒攻击的测试损失

Fig.8　Test loss against data poisoning attacks in long-tail data scenario

6　结语

本文基于被动方不可信的安全假设,设计了一种纵向联邦学习中的安全加权聚合方案。具体而言,为了在长尾数据场景下抵御被动方发动的数据投毒攻击,基于效用评估设计一种有毒数据过滤算法,并且根据效用评估结果等指标来自适应的为嵌入向量分配聚合权重,同时基于 SHE 和掩蔽机制来保证嵌入向量在损失计算以及权重计算等阶段的隐私性。实验结果表明,本文方案在普通场景和长尾数据场景中都可以有效抵御被动方发动的投毒攻击,模型收敛率显著高于其他两个对比方案。未来将重点提升方案的效率,进一步降低各方在协同进行损失计算时的通信轮数。

参考文献:

[1] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data

[C]//Proceedings of Artificial Intelligence and Statistics. Cambridge: PMLR, 2017:1273-1282.
[2] ROMANINI D, HALL A J, PAPADOPOULOS P, et al. Pyvertical: a vertical federated learning framework for multi-headed splitnn[EB/OL]. arXiv: https://arxiv.org/abs/2104.00489.
[3] LUO X J, WU Y C, XIAO X K, et al. Feature inference attack on model predictions in vertical federated learning[C]//2021 IEEE 37th International Conference on Data Engineering (ICDE). Chania: IEEE, 2021:181-192.
[4] ERDOĞAN E, KÜPÇÜA, ÇIÇEK A E. UnSplit: data-oblivious model inversion, model stealing, and label inference attacks against split learning[C]//Proceedings of the 21st Workshop on Privacy in the Electronic Society. Los Angeles: ACM, 2022: 115-124.
[5] LIU Z L, CHEN Y Y, YU H, et al. GTG-shapley: efficient and accurate participant contribution evaluation in federated learning[J]. ACM Transactions on Intelligent Systems and Technology, 2022, 13(4):1-21.
[6] 王勇,李国良,李开宇. 联邦学习贡献评估综述[J]. 软件学报,2023,34(3):1168-1192.
WANG Yong, LI Guoliang, LI Kaiyu. Survey on contribution evaluation for federated learning [J]. Journal of Software, 2023, 34 (3):1168-1192.
[7] BONAWITZ K, IVANOV V, KREUTER B, et al. Practical secure aggregation for privacy-preserving machine learning[C]// Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas: ACM, 2017:1175-1191.
[8] MAHDIKHANI H, LU R X, ZHENG Y D, et al. Achieving $O(\log^3 n)$ communication-efficient privacy-preserving range query in fog-based IoT[J]. IEEE Internet of Things Journal, 2020, 7(6):5220-5232.
[9] SHEN S Q, TOPLE S, SAXENA P. Auror: defending against poisoning attacks in collaborative deep learning systems[C]// Proceedings of the 32nd Annual Conference on Computer Security Applications. Los Angeles California: ACM, 2016:508-519.
[10] FUNG C, YOON C J M, BESCHASTNIKH I. The limitations of federated learning in sybil settings [C]//Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020). 2020:301-316.
[11] ANDREINA S, MARSON G A, MOLLERING H, et al. BaFFLe: backdoor detection via feedback-based federated learning [C]//2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS). Washingtom: IEEE, 2021: 852-863.
[12] ZHAO L C, WANG Q, ZOU Q, et al. Privacy-preserving collaborative deep learning with unreliable participants[J]. IEEE Transactions on Information Forensics and Security, 2019, 15:1486-1500.
[13] QIU P, ZHANG X, JI S, et al. Hijack vertical federated learning models with adversarial embedding[EB/OL]. arXiv: https://arxiv.org/abs/2212.00322.
[14] HE Y, SHEN Z L, HUA J Y, et al. Backdoor attack against split neural network-based vertical federated learning[J]. IEEE Transactions on Information Forensics and Security, 2023, 19:748-763.
[15] WANG S, GAI K K, YU J, et al. BDVFL: blockchain-based decentralized vertical federated learning[C]//2023 IEEE International Conference on Data Mining (ICDM). Shanghai: IEEE, 2023:628-637.
[16] GAO X, ZHANG L. PCAT: functionality and data stealing from split learning by pseudo-client attack[C]//Proceedings of the 32nd USENIX Security Symposium, 2023:5271-5288.
[17] SATHYA S S, VEPAKOMMA P, RASKAR R, et al. A review of homomorphic encryption libraries for secure computation [EB/OL]. arXiv: https://arxiv.org/abs/1812.02428.
[18] BOYLE E, GILBOA N, ISHAI Y. Function secret sharing: improvements and extensions[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna: ACM, 2016:1292-1303.
[19] HUANG Y M, WANG W W, ZHAO X Y, et al. EFMVFL: an efficient and flexible multi-party vertical federated learning without a third party[J]. ACM Transactions on Knowledge Discovery from Data, 2024, 18(3):1-20.
[20] CAI S W, CHAI D, YANG L, et al. Secure forward aggregation for vertical federated neural networks[M]//Trustworthy Federated Learning. Cham: Springer, 2023:115-129.
[21] FU F C, XUE H R, CHENG Y, et al. BlindFL: vertical federated machine learning without peeking into your data[C]// Proceedings of the 2022 International Conference on Management of Data. Philadelphia: ACM, 2022:1316-1330.
[22] SUN H, ZHANG Y, LI M X, et al. FLFHNN: an efficient and flexible vertical federated learning framework for heterogeneous neural network[M]//Wireless Algorithms, Systems, and Applications. Cham: Springer Nature, 2022:338-350.
[23] CHEN T Y, JIN X, SUN Y J, et al. Vertical asynchronous federated learning: algorithms and theoretic guarantees[M]// Federated Learning. Amsterdam: Elsevier, 2024:199-217.
[24] THAPA C, MAHAWAGA ARACHCHIGE P C, CAMTEPE S, et al. SplitFed: when federated learning meets split learning

[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(8):8485-8493.
[25] XU D P, YUAN S H, WU X T. Achieving differential privacy in vertically partitioned multiparty learning[C]//2021 IEEE International Conference on Big Data (Big Data). Orlando: IEEE, 2021:5474-5483.
[26] SHI H R, XU Y H, JIANG Y L, et al. Efficient asynchronous multi-participant vertical federated learning[J]. IEEE Transactions on Big Data, 2024, 10(6):940-952.
[27] LI S, YAO D, LIU J. FedVS: straggler-resilient and privacy-preserving vertical federated learning for split models[C]// Proceedings of the International Conference on Machine Learning. Cambridge: PMLR, 2023:20296-20311.
[28] WANG S, GAI K, YU J, et al. VFedMH: vertical federated learning for training multi-party heterogeneous models[J]. [2024-10-15] https://arxiv.org/abs/2310.13367.
[29] MISHRA P, LEHMKUHL R, SRINIVASAN A, et al. Delphi: a cryptographic inference service for neural networks[C]// Proceedings of the 29th USENIX Security Symposium. Los Alamitos: IEEE, 2020:2505-2522.
[30] XIA W S, LI Y, ZHANG L, et al. Cascade vertical federated learning towards straggler mitigation and label privacy over distributed labels[J]. IEEE Transactions on Big Data, 2024, 10(6):926-939.

(编辑:祁业卿)

(上接第 28 页)

[22] GUO Daya, REN Shuo, LU Shuai, et al. GraphCodeBERT: pre-training code representations with data flow[C]//2021 International Conference on Learning Representations. ICLR, 2021.
[23] WU Hongjun, et al. Peculiar: smart contract vulnerability detection based on crucial data flow graph and pre-training techniques[C]//2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2021.
[24] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains[C]//2018 13th EuroSys Conference. New York: ACM, 2018:1-15.
[25] SOUSA J, BESSANI A, VUKOLIC M. A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform[C]//2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2018.
[26] HUANG Yongfeng, BIAN Yiyang, et al. Smart contract security: a software lifecycle perspective[J]. IEEE Access, 2019, 7:150184-150202.
[27] LV Penghui. Potential risk detection system of hyperledger fabric smart contract based on static analysis[C]//2021 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2021:1-7.
[28] YAMASHITA K, NOMURA Y, ZHOU E, et al. Potential risks of hyperledger fabric smart contracts[C]//2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE). IEEE, 2019:1-10.
[29] LI Peiru, WANG Yizheng, HUANG Hao, et al. A vulnerability detection framework for hyperledger fabric smart contracts based on dynamic and static analysis[C]//Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering. New York: ACM, 2022:366-374.
[30] XU Xiaofei, HU Tiaoyuan, LI Bixin, et al. CCDetector: detect chaincode vulnerabilities based on knowledge graph[C]// 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2023:699-704.
[31] LUA T. Tree-sitter[EB/OL]. 2023. https://tree-sitter.github.io/tree-sitter/.

(编辑:祁业卿)