成绩：

计算机与网络安全
（必修）

课程论文

论文题目 云计算中的 AI 与安全多方计算：挑战与机遇

学生姓名 盖 乐 学号 21009200991

Email 571753112@qq.com 电话 15702996236

2024 年 6 月 6 日

云计算中的 AI 与安全多方计算：挑战与机遇

摘　要：深度学习技术的迅猛发展推动了大规模神经网络模型在自然语言处理等领域的广泛应用，其中以 GPT 为代表的大模型推理引起了学术界和产业界的广泛关注。然而，这些强大模型在云计算过程中可能引发潜在的隐私泄露问题，构成了严重的安全挑战。本文全面回顾了当前隐私机器学习领域的研究进展，详细探讨了安全多方计算的定义、分类以及相关的密码学基础，如秘密共享、混淆电路、同态加密和不经意传输。本文对 19 个主流框架进行了系统的调查和分类，着重分析了各框架所采用的密码学技术，深入剖析了它们的优缺点。此外，本文对最新的研究内容进行了全面总结和概括，特别关注了模型压缩加速与安全多方计算的融合，该结合在提高隐私机器学习效率方面取得了显著的成果，尤其针对大模型的推理。

最后，通过分析安全多方学习与其他隐私保护机器学习技术的差异，本文勾勒了安全多方计算在隐私机器学习领域未来的发展方向。这一综合而深入的调研为推动隐私保护技术的进步提供了重要的参考和指导。

关键词：隐私计算，隐私机器学习，安全多方计算，同态加密，云计算

AI and secure multi-party computing in cloud computing: challenges and opportunities

Abstract: The rapid development of deep learning technology has driven the widespread application of large-scale neural network models in fields like natural language processing, with GPT-like large models' inference garnering widespread attention in both academia and industry. However, these powerful models may pose potential privacy leakage issues during inference, presenting serious security challenges. This paper provides a comprehensive review of current research progress in privacy-preserving machine learning, exploring in detail the definitions, classifications, and cryptographic foundations of secure multiparty computation, including secret sharing, garbled circuits, homomorphic encryption, and oblivious transfer. It systematically investigates and categorizes 19 mainstream frameworks, focusing on the cryptographic techniques they employ and thoroughly analyzing their strengths and weaknesses. Additionally, the paper comprehensively summarizes and generalizes the latest research findings, particularly emphasizing the integration of model compression acceleration and secure multiparty computation, which has achieved significant results in improving the efficiency of privacy-preserving machine learning, especially in large model inference.

Finally, by analyzing the differences between secure multiparty learning and other privacy-preserving machine learning techniques, this paper outlines future development directions for secure multiparty computation in the field of privacy-preserving machine learning. This comprehensive and in-depth survey provides crucial references and guidance for advancing privacy protection technologies.

Keyword: Privacy Computing, Privacy-preserving Machine Learning, Secure Multiparty Computation, Homomorphic Encryption, Secret Sharing, Garbled Circuits

1 引言

近年来，随着全球移动互联网的普及和数字经济及大数据产业的蓬勃繁荣，全球数据总量增长迅速，数据已经成为重要的生产要素之一。这一变化对以土地、劳动和资本等“硬要素”为基础的传统市场机制产生了深刻影响[41,42]。随着新技术的不断涌现，如数据处理、数据分析和数据挖掘等，数据已经深度融入生产过程，并推动各行各业的产业升级。例如，金融领域利用相关数据进行风险评估，零售领域则通过大数据联合营销、个性化推荐等方式增强市场竞争力，医疗领域则利用数据实现智能医疗等。数据作为新型生产要素，是我国经济社会发展的基础资源和创新引擎。

表格 1 数据要素市场化发展政策

时间	政策	意义
2020 年 3 月	《中共中央国务院关于构建更加完善的要素市场化配置体制机制的意见》	强调了发展数据要素市场的重要性
2020 年 5 月	《中共中央国务院关于新时代加快完善社会主义市场经济体制的意见》	强调了数据要素市场是社会主义市场经济的重要组成部分
2021 年 3 月	《中华人民共和国国民经济和社会发展第十四个五年规划和 2035 年远景目标纲要》	提到要完善健全数据要素流通的标准和保护数据安全
2021 年 12 月	《要素市场化配置综合改革试点总体方案》	研究建立数据要素流通规则
2022 年 12 月	《中共中央国务院关于构建数据基础制度更好发挥数据要素作用的意见》	提出要建立安全可控、弹性包容的数据要素治理制度等

如表 1 所示，近年来政府不断发布相关政策，进一步加强对数据要素的研究和管理。这些政策为未来数据产业的发展提供了有力支持和保障。在政策的支持下，云计算技术应用逐渐扩大，涉及多个领域。云计算通过提供弹性、可扩展的计算资源，促进了数据的处理和存储，为企业提供了更灵活、高效的解决方案。然而，随着云计算的广泛应用，数据泄露问题逐渐凸显。隐私泄露问题成为云计算领域的重大挑战。由于数据跨越多个网络和存储系统，可能存在计算过程中泄露的风险。在这方面，政府也需要加强对云计算环境下数据安全的监管和管理，确保数据在传输，存储和计算过程中得到妥善保护。

因此，解决云计算环境下的隐私泄露问题需要更完善、更有效的技术手段。隐私计算技术是一种备受关注的技术，它在保证数据安全的前提下进行数据分析和运算，同时不泄露任何相关信息。目前，隐私计算技术主要包括联邦学习、安全多方计算和可信执行环境。其中，联邦学习的安全性主要建立在差分等技术上，会泄露部分梯度；可信执行环境则是一种针对硬件的可信计算技术；安全多方计算则基于密码学协议，具有更高的安全性。在云计算环境下，采用这些技术可以

更好地保护用户的隐私信息，防止数据泄露。

1.1 相关综述

近年来，研究人员对隐私保护机器学习技术进行了调查。Hastings 等人[15]评估了 11 个安全多方计算的通用编译器，关注可用性、示例程序、功能性、实施标准等五个方面，但未涵盖编译器与机器学习的整合，且专注于通用编译器而忽略了对安全多方学习框架的研究。Lindell 等人[23]介绍了安全多方计算，但未扩展到隐私机器学习领域。Lushan Song 等人[33]总结了近期隐私技术，但未对隐私机器学习的交叉领域进行总结。Pastrana 等人[3]提供了理论背景，但未引入交叉学科以推动隐私计算。蒋瀚等人[19]综述了隐私保护机器学习的密码学方法，但对安全多方学习框架的研究不够全面，尤其是缺乏对大模型框架隐私推理分析的调查。此外，所分析的安全多方学习框架特征较少，缺少底层安全多方计算技术和支持的机器学习类型等特征。

1.2 本文贡献

近年来，在安全多方计算协议的支持下，研究人员设计了众多安全多方学习框架。本文全面调研了现有安全多方学习技术及框架，主要贡献如下：

- 详细介绍了安全多方计算的概念和机器学习知识，全面分析了安全多方计算采用的安全模型和场景。
- 根据安全多方学习框架底层依赖的原语和隐私保护技术，对最近提出的 19 个框架进行分类，并深入分析它们的优势和局限性。
- 在基础框架的基础上，对新领域的交叉进行了分析，特别关注模型压缩对隐私计算的加速效果，以及隐私计算在大模型推理方面的保驾护航。
- 分析了安全多方学习与其他隐私保护机器学习技术的区别，并对安全多方计算在隐私机器学习的应用提出了未来发展建议。

2 安全定义

本节简要介绍了安全多方计算的基本概念和分类标准，并根据分类介绍了相关的安全性证明原理。

2.1 安全多方计算概念

安全多方计算（Secure Multi-party Computation，简称 MPC）是一种分布式合作计算任意函数的算法，它能够保证不泄露参与方的隐私输入和输出，被认为是加密的分布式计算。该算法最初由姚期智[38]于 1982 年在 FOCS 学术会议上提出。

MPC 的主要目标是在一组互不信任的参与方之间联合完成计算，保证每个参与方都能得到正确的运算结果，并且不会泄露其他的数据和运算结果外的任何信息。假设有 n 个参与方 $P_1, P_2, \ldots, P_n$，他们各自持有输入数据 $x_1, x_2, \ldots, x_n$。在不泄露输入信息的前提下，这 n 个参与方联合完成函数 $f(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$ 的计算。通过 MPC 算法，每个参与方都能得到协议规定的正确运算结果。

为了正式声明和证明协议是安全的，需要对多方计算的安全性进行精确定义。

安全多方计算的安全特征主要包括计算隐私性、正确性、公平性、输入独立性和输出可达性等[9]。

1. 隐私性:任何一方都不能获得超过协议规定输出的信息。
2. 正确性:任何一方都可以获得协议保证的正确的输出。
3. 公平性:当且仅当诚实的参与方也收到其输出时，腐败的参与方应收到其输出。
4. 输入的独立性:输入之间相互独立。腐坏方须独立于诚实的参与方的输入来选择其输入。
5. 输出的可达性:当存在腐败的参与方时，在协议结束后仍然能够获得预期的计算结果。

在考虑安全性时，不能忽略攻击协议执行的对手力量。定义参与协议的破坏方为腐败方，为协议的攻击者。根据允许的对抗行为，安全多方计算协议可以分为半诚实的对手（腐败方也会正确地遵循协议规范），恶意的对手（腐败方可以根据对手的指示任意偏离协议规范），隐蔽的对手（对手和腐败方可能会恶意破坏协议）。根据腐败策略可以将安全多方计算协议分为静态腐败模型，自适应腐败模型，主动安全模型。此外根据诚实方和腐败方数量可以分为诚实大多数和恶意大多数。

2.2 安全多方计算安全模型

安全多方计算的目的是让一组参与者联合学习其敏感输入的某些商定功能的正确输出，而不透露任何其他信息。本节提供了一个更正式的定义，以阐明安全多方计算旨在提供的安全性。首先，本文介绍真正的理想范式，它构成了定义安全的概念核心。然后介绍了安全多方计算常用的两种不同的对手模型。最后，介绍组合问题，即当安全协议调用另一个子协议时，安全性是否以自然方式保持。

1. Real-Ideal 范式

理想世界:在理想世界，假设有一个可信的第三方$\mathcal{T}$。参与方P_i与可信第三方可以安全的传输信息，将自己的私有数据x_i发送给$\mathcal{T}$。可信第三方$\mathcal{T}$独自完成计算$\mathcal{F}(x_1,x_2,\ldots,x_n)$的计算。最终，可信第三方$\mathcal{T}$将计算结果$(y_1,y_2,\ldots,y_n)$发送到各个参与方，可信第三方不会被腐败，保证结果的运算正确。参与者相互独立，只能看到自己的输入和输出。

现实世界:在理想世界，可信的第三方的可能存在性很低。参与者需要通过安全协议进行交互，来实现理想世界的功能函数，完成计算目标。当然现实世界会发生参与者腐败的情况，这种情况下达到的攻击效果应该和理想世界一致。

简单来说，在 Real-Ideal 范式中，敌人在现实世界的攻击和在理想世界的攻击效果一样，不能分别出来，就可以证明现实世界中协议是安全的。

2. 半诚实安全性

半诚实敌手(Semi-honest Adversary)指的是虽然会腐败部分参与者，但仍然会遵

守协议规则执行协议。下面用 Real-Ideal 范式来表示半诚实安全性。

设π是一个协议，$\mathcal{F}$是一个功能函数。令$\mathcal{C}$为被腐败的参与方， 假设Sim是模拟器算法。本文定义了以下随机变量分布：

- $Real_{\pi}(\kappa,\mathcal{C};x_1,\ldots,x_n)$\$:在安全参数κ下，每个参与方P_i输入x_i执行协议。令V_i为参与方P_i最终的视角，y_i为最终的输出。协议输出为$\{V_i|\ i\in\mathcal{C}\},(y_1,\ldots,y_n\}$。
- $Ideal_{\mathcal{F},Sim}(\kappa,\mathcal{C};x_1,\ldots,x_n)$计算$\mathcal{F}(x_1,\ldots,x_n)\to(y_1,\ldots,y_n)$，协议输出$Sim(\mathcal{C},\{(x_i,y_i)|\ i\in C\}),(y_1,\ldots,y_n)$。

如果现实世界和理想世界各方的视角和输出不可区分，即$Real_{\pi}(\kappa,\mathcal{C};x_1,\ldots,x_n)$和$Ideal_{\mathcal{F},Sim}(\kappa,\mathcal{C};x_1,\ldots,x_n)$在$\kappa$下概率分布不可区分，那么就可以证明该协议是安全的。

3. 恶意安全性

恶意攻击者(Malicious)又称为主动攻击者，在协议执行过程中，可以控制腐败的参与方任意偏离协议规则执行协议。相对于半诚实安全，恶意攻击者可以控制网络，破坏正常的协议输出和输入。这就需要协议有更强的安全性来应对这种攻击，在协议进行过程中及时发现攻击并终止协议。下面用 Real-Ideal 范式来表示恶意安全性。

设$\mathcal{A}$是一个攻击者，$corrupt(\mathcal{A})$表示现实世界中被腐败的参与方集合。$corrupt(Sim)$表示理想世界中被腐败的参与方集合。本文定义了以下随机变量分布：

- $Real_{\pi,\mathcal{A}}(\kappa;\{x_i|\ i\notin corrupt(\mathcal{A})\})$:在安全参数$\kappa$下，每个诚实的参与方$P_i$输入$x_i$执行协议，而腐败的参与方按照攻击者的指示进行输入。令V_i为参与方P_i最终的视角，y_i为诚实参与方的输出。协议输出为$(\{V_i|\ i\in corrupt(\mathcal{A})\},\{y_i|\ i\notin \text{corrupt}(\mathcal{A})\})$。
- $Ideal_{\mathcal{F},Sim}(\kappa;\{x_i|\ i\notin corrupt(\mathcal{A})\})$:执行$Sim$,直到输出一个输入集合$\{x_i|\ i\in corrupt(\mathcal{A})\}$。计算输出$\mathcal{F}(x_1,\ldots,x_n)\to(y_1,\ldots,y_n)$,协议将输出$\{y_i|\ i\in corrupt(\mathcal{A})\}$发送到$Sim$。$V^*$表示$Sim$最终的输出。输出

$(V^*, \{y_i | \ i \notin corrupt(Sim)\})$。

如果对于现实世界的攻击者 $\mathcal{A}$，存在 $corrupt(\mathcal{A}) = corrupt(Sim)$，使得现实世界和理想世界诚实参与者的视角和输出不可区分，即 $Real_{\pi,\mathcal{A}}(\kappa; \{x_i | \ i \notin corrupt(\mathcal{A})\})$ 和 $Ideal_{\mathcal{F},Sim}(\kappa; \{x_i | \ i \notin corrupt(\mathcal{A})\})$ 在 κ 下概率分布不可区分，那么就可以证明该协议在恶意攻击下是安全的。

这里协议一般还会包括一些其他模块，如交互功能函数等，用来判断协议是否已经发生错误，在关键的适合进行中断，称为可中止安全性(Security with Abort)。

4. 混合世界与组合安全

为了更方便的使用各种协议，在设计协议时通常会采用模块化设计。在设计协议的过程中调用其他的理想功能函数进行实现，这就需要安全模型具有组合性。保证组合性的标准方式是使用 Canetti[4]提出的通用可组合性 UC 框架。UC 框架在之前的安全模型上进行扩展，增加了一个称为环境的部分，接下来用 Real-Ideal 范式来表示 UC 的组合安全性。

设 $\mathcal{Z}$ 是一个环境的实体:

- $Real_{\pi,\mathcal{A},\mathcal{Z}}(\kappa)$:攻击者 $\mathcal{A}$ 和环境 $\mathcal{Z}$ 的协议交互过程。
- $Ideal_{\mathcal{F},Sim,\mathcal{Z}}(\kappa)$:攻击者 Sim 和环境 $\mathcal{Z}$ 的协议交互过程。

对于给定协议 π，如果对于现实世界的攻击者 $\mathcal{A}$，存在 $corrupt(\mathcal{A}) = corrupt(Sim)$ 的仿真者 Sim，对于所有环境实体 $\mathcal{Z}$:

$$|\Pr[Real_{\pi,\mathcal{A},\mathcal{Z}}(\kappa) = 1] - \Pr[ldeal_{\mathcal{F},Sim,\mathcal{Z}}(\kappa) = 1]|.$$

如果在 κ 下可以忽略，可以证明这个协议 UC-安全实现了 $\mathcal{F}$。

3 底层原语和机器学习技术

本节介绍隐私计算中常用的密码学基础知识，主要包括秘密共享，混淆电路和不经意传输和机器学习相关知识。这些密码学原语是密码学协议的基础，大部分为用 c++实现并集成在协议底层，是安全多方计算不可或缺的一部分。

3.1 秘密共享技术

秘密分享（Secret Sharing，简称 SS）通过将秘密分成 n 部分，并在 n 个参与者中分享，只有当参与者数量超过特定的 t 个时，才能够确定或恢复秘密，而在少于 t 个参与者的情况下则无法获得任何有关秘密的信息。秘密共享是安全多方计算的核心，最早由 Sharmir 和 Blakley 在 1979 年提出。Sharmir 基于 Lagrange 插值多项式，Blakley 是基于线性几何投影理论的。

秘密共享的方案通常分为秘密分享和秘密重构两个步骤。

1. 秘密分享

假设参与者分别是$C=\{C_1,C_2....,C_n\}$，参与者之间共享一个秘密S，秘密拥有者将S作为秘密分享算法$ShareSecret(S)$的输入，输出每个参与者的秘密份额$(s_1,s_2,\ldots,s_n)$。

$$ShareSecret(S)\rightarrow(s_1,s_2,\ldots,s_n).$$

2. 秘密重构

只有当参与者数量超过特定的 t 个时，才能够重构秘密，假设有 t 个参与者，分别是$C=\{C_1,C_2....,C_t\}$,通过调用秘密重构算法$Recon()$来恢复秘密S。

$$Recon(s_1,s_2,\ldots,s_t)\rightarrow S.$$

Shamir 利用拉格朗日插值法构建了经典的(t,n)门限秘密共享方案，算法思路如下：随机选择一个素数 p，并产生一个随机的$t-1$次多项式：$f(x)=a_{t-1}x^{t-1}+\ldots+a_1x+a_0 \mod p$。其中令$a_0$为秘密$s$，即$f(0)=a_0=s$，在秘密分享步骤中，随机选择 n 个互不相同的整数$x_1,x_2,\ldots,x_n$，计算$S_i=f(x_i)$并将s_i作为秘密共享份额分别发给 n 个参与者P_i。重构秘密时，n 个参与方中的任意 t 个参与方将秘密共享份额聚在一起根据拉格朗日插值法即可重构出秘密 S。

根据几何中多维空间点的性质，Blakley 从另一种角度提出了(t,n)门限秘密共享方案。把 t 维空间上的一个点看作秘密 s，每个秘密份额s_i为这个点的 t-1 维超平面，t 维空间上的点 s 可以被任意一个 t-1 维超平面确定，故可以重构秘密 s。

3.2 不经意传输技术

不经意传输（OT, Oblivious Transfer）是密码学中的一种基本协议，用于实现安全的两方通信，在安全多方计算等领域得到了广泛应用[43]。在该协议中，发送方和接收方进行通信，执行结束后，接收方能够获取所需信息，而不会泄漏其他信息给发送方，同时，发送方也不知道接收方获取了哪个信息。最著名的 OT 协议是由 Even 等人提出的 1-out-of-2 协议，它使用公钥密码体制，提供了 OT 的公理化定义和实现方式。具体来说，Alice 拥有两个秘密m_0和m_1，而 Bob 想要知道其中一个。在 OT 协议完成后，Bob 可以获取其中一个秘密，但不能知道另一个秘密是什么，同时 Alice 也不知道 Bob 选择的是哪一个，m_0还是m_1。

具体 OT 协议参与方如下：

- 发送方$\mathcal{S}$输入秘密$m_1,m_2\in\{0,1\}^n$。

- 接收方 $\mathcal{R}$ 输入选择 $b \in \{0,1\}$。

OT 协议过程：

1. $\mathcal{R}$ 随机生成公私钥 (sk, pk),并生成一个随机公钥 pk'。如果选择 $b=0$, $\mathcal{R}$ 将 (pk, pk') 发送给 $\mathcal{S}$，否则 $b=1$, $\mathcal{R}$ 将 (pk', pk) 发送给 $\mathcal{S}$。
2. $\mathcal{S}$ 接收 (pk_0, pk_1) 并向 $\mathcal{R}$ 发送密文 $(e_1, e_2) = (ENC_{pk_0}(m_0), ENC_{pk_1}(m_1))$。
3. $\mathcal{R}$ 接收密文 (e_0, e_1),并用 sk 解密密文 e_b。

在 1986 年之后，Brassard 等人进一步改进了 OT 协议，将其扩展到了 1-out-of-n OT 版本。这个版本与之前提到的 1-out-of-2 OT 基本相同，唯一不同的是现在需要传递 n 条秘密给 Bob，而不是之前的两条。后续逐渐完善和发展，产生了 ROT，COT 等变形协议，并且从俩方交互逐渐变成多方交互。

3.3 混淆电路技术

混淆电路是姚期智教授在解决姚氏百万富翁问题时提出的密码学协议可以说是本文的基础。众所周知，计算机的底层是 0-1 的二进制电路运算，混淆电路就是针对计算电路进行混淆。逻辑电路可以通过 2 方的输入输出 0-1，构建复杂的电路来计算。设功能函数 $\mathcal{F}$ 表示为电路 $\mathcal{C}$，在两方的混淆电路中，P_1 生成密钥并加密电路，P_2 在未知密钥和明文关系的情况下解密输出。步骤如下:

P_1 为电路 $\mathcal{C}$ 的导线 w_i 指定俩个密钥 k_i^0 和 k_i^1，俩个密钥分别与导线的俩个明文值相关。针对布尔电路 $\mathcal{C}$，对于每个门 G 的输入导线 w_i，w_j 和输出导线 w_t，P_1 构建如下加密电路:

$$T_G = \begin{pmatrix} \mathrm{Enc}_{k_i^0, k_j^0}\left(k_t^{G(0,0)}\right) \\ \mathrm{Enc}_{k_i^0, k_j^1}\left(k_t^{G(0,1)}\right) \\ \mathrm{Enc}_{k_i^1, k_j^0}\left(k_t^{G(1,0)}\right) \\ \mathrm{Enc}_{k_i^1, k_j^1}\left(k_t^{G(1,1)}\right) \end{pmatrix}$$

如果 G 是一个 AND 门，对应加密电路为:

$$T_G = \begin{pmatrix} \mathrm{Enc}_{k_i^0,k_j^0}\left(k_t^0\right) \\ \mathrm{Enc}_{k_i^0,k_j^1}\left(k_t^0\right) \\ \mathrm{Enc}_{k_i^1,k_j^0}\left(k_t^0\right) \\ \mathrm{Enc}_{k_i^1,k_j^1}\left(k_t^1\right) \end{pmatrix}$$

然后，p_1 将上面的加密电路进行打乱，构建混淆电路。P_2 通过不经意传输协议获取到激活标签，根据混淆电路对电路求解得到与输出导线相关的密钥。P_2 将得到的密钥发生给 P_1 进行解密，完成功能函数 $\mathcal{F}$ 的安全求解。

3.4 机器学习基础知识

在本小节简要解释了基本机器学习算法：线性回归、卷积神经网络。这里所涉及的算法都是经典的算法，可以在标准的机器学习和深度学习教科书中找到。

1. 线性回归

给定 n 个训练数据样本 x_i，每个样本包含 d 个特征和对应的输出标签 y_i，回归是学习函数 g 以使得 $g(X_i) \approx y_i$ 的统计过程。回归在现实生活中有很多应用。

在线性回归中，假设函数 g 是线性的，并且可以表示为 X_i 与系数向量 W 的内积:

$$g(X_i) = \sum_{j=1}^{d} x_{ij} w_j = X_i \cdot W_j.$$

为了学习系数向量 W，定义了一个损失函数 $C(W)$，并通过优化的 $argmin_w C(W)$ 来计算 W。在线性回归中，常用的成本函数是 $C(W) = \frac{1}{n}\sum C_i(W)$，其中 $C_i(W) = \frac{1}{2}(X_i \cdot W ? y_i)^2$。

2. 随机梯度下降

SGD 算法（Stochastic Gradient Descent）是一种常见的优化算法，它是一种基于梯度下降的逼近算法，可以在大规模数据集上有效地进行模型优化。在机器学习领域中，SGD 算法被广泛应用于线性回归、逻辑回归和神经网络等模型的训练中。

SGD 算法的工作原理如下：W 被初始化为随机值的向量或全为 0。在每次迭代中，随机选择样本 (x_i, y_i) 并且将系数 w_j 更新为:

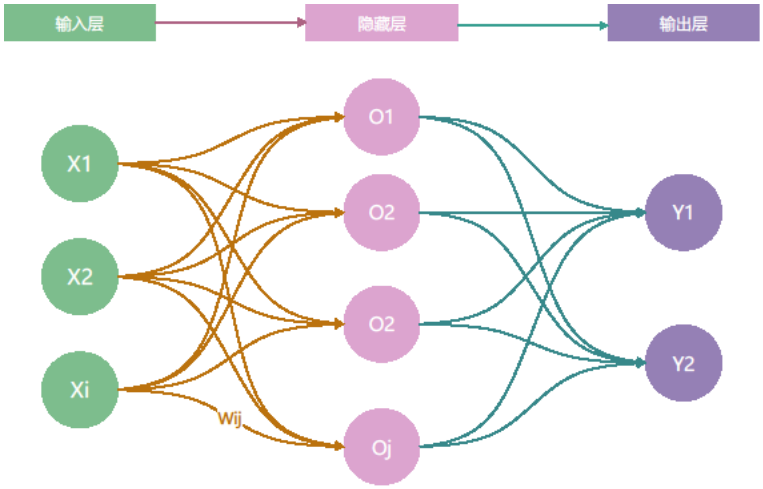
$$w_j := w_j - \alpha \frac{\partial C_i(\mathbf{w})}{\partial w_j}.$$

其中，α 是学习率，用于控制每次更新的步长。代入线性回归的成本函数，公式为 $w_j := w_j - \alpha(X_i \cdot W - y_i)x_{ij}$。

3. 神经网络

神经网络是一种受生物神经系统启发的人工智能技术。它们模拟人脑中神经元的连接和活动，以执行复杂的计算任务。神经网络的核心思想是通过多层神经元对大量输入数据进行处理和转换，产生一个输出结果。

神经网络由多个神经元组成，每个神经元接收多个输入信号。使用激活函数对这些信号进行加权和计算以产生输出信号。多个神经元按一定的拓扑结构排列形成神经网络，通常包括输入层、隐藏层和输出层。如图 1 所示，输入层接收原始数据，隐藏层对数据进行非线性变换和抽象，输出层将隐藏层的结果映射到目标空间。



图表 1 神经网络示意图

神经网络的训练通常使用反向传播算法，该算法从网络输出开始，向后传播误差以更新神经元之间的权重，使网络输出与实际结果之间的差异最小化。通过迭代训练，神经网络可以逐渐学习输入数据的特征并产生更准确的输出结果。

形式上，神经网络可以表示为将输入 x 映射到输出 y 的数学函数 f：

$$y = f(x).$$

函数 f 由多层神经元组成，每一层都可以看成和逻辑回归类似的结构 g：

$$f(x) = (L_n \cdot g(L_2(g(L_1(x))))).$$

其中 $L_1, L_2, \cdot, L_n$ 是神经元的线性层， g 是应用于每个神经元输出的激活函数。

设神经网络的训练使用的最小化损失函数为 E，它衡量网络输出与实际结果之

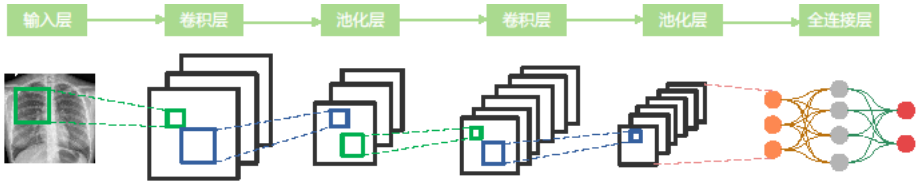
间的差异：

$$E = L(y, y').$$

其中 y' 是实际输出，L 是量化 y 和 y' 之差的损失函数。反向传播用于计算损失函数相对于网络权重的梯度，用于通过随机梯度下降等优化算法更新权重。

5. 卷积网络

卷积神经网络（Convolutional Neural Network，简称 CNN）是一种广泛应用于图像、视频、语音等领域的深度学习模型。它的设计灵感来源于生物学中视觉皮层的结构和功能，通过多层卷积和池化操作对输入数据进行特征提取和降维，并通过全连接层进行分类或回归。



图表 2 卷积神经网络结构图

如图 2 所示 CNN 的基本结构由卷积层、激活函数、池化层和全连接层等组成。

卷积层(Convolutional Layer)：卷积层通过在输入数据的局部区域上应用卷积操作，提取出局部和全局的特征。卷积操作可以看作是一种滤波器，将输入数据和一个可训练的卷积核进行逐元素乘积和求和，从而得到一个新的特征映射。每个卷积核可以捕捉到输入数据中的一种特定模式或特征。通常，卷积层会应用多个卷积核，并且每个卷积核都可以学习到不同的特征，这些特征在后续的层次结构中被用于进一步的特征提取和分类。卷积层的数学表达式可以写为:

$$f_i = \sigma\left(\sum_{j=1}^{m} k_j \cdot x_{i+j-1} + b\right).$$

激活函数(Activation Function)：激活函数用于在卷积层之后引入非线性变换。常用的激活函数包括 ReLU（Rectified Linear Unit）、sigmoid、tanh 等。常见的激活函数包括:

- ReLU 函数: $f(x) = \max(0, x)$ 。
- Sigmoid 函数: $f(x) = \dfrac{1}{1+e^{-x}}$ 。
- tanh 函数: $f(x) = \dfrac{1-e^{-2x}}{1+e^{-2x}}$ 。

池化层(Pooling Layer)：池化层是 CNN 中的另一个重要层次结构，用于减少特征映射的空间尺寸和数量，从而降低网络的计算复杂度和参数数量，增强网络的鲁棒性和泛化能力。常见的池化操作包括最大池化和平均池化，它们分别取特征

映射中局部区域的最大或平均值作为输出值。通过对特征映射的下采样，池化层可以使得网络更加关注于输入数据的重要特征，并且减少了对输入数据的局部变化的敏感性。

最大池化操作的数学表达式为:

$$f_i = \max_{j=1}^{m} x_{i+j-1}.$$

平均池化操作的数学表达式为:

$$f_i = \frac{1}{m}\sum_{j=1}^{m} x_{i+j-1}.$$

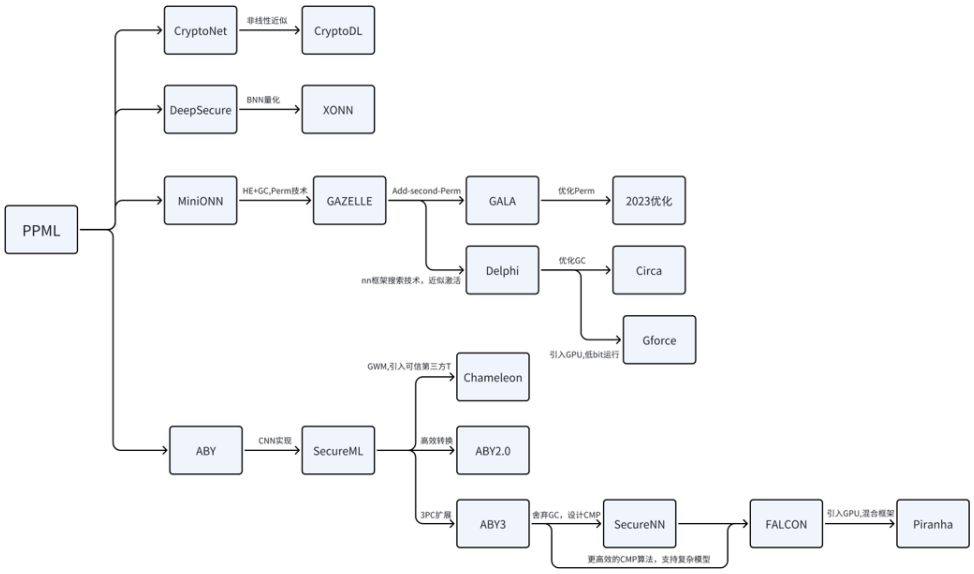
全连接层（Fully Connected Layer）}：全连接层是 CNN 中的最后一个层次结构，它将卷积层和池化层中提取的特征映射转化为分类或回归结果。在全连接层中，每个神经元与前一层中的所有神经元相连接，并且每个神经元都有自己的权重和偏置项。在分类问题中，全连接层通常使用 softmax 函数作为输出激活函数，用于将网络的输出映射到类别概率空间中。在回归问题中，全连接层通常使用线性全连接层的数学表达式可以写为:

$$f = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right),$$

其中 x_i 是前一层中的第 i 个神经元输出， w_i 是该层中与第 i 个神经元相连的权重，b 是偏置项，σ 是激活函数。全连接层将前一层中提取的特征映射转化为分类或回归结果。在分类问题中，全连接层通常使用 softmax 函数作为输出激活函数，用于将网络的输出映射到类别概率空间中。在回归问题中，全连接层通常使用线性函数作为输出激活函数。

4 隐私机器学习框架

如图 3 所示，本章根据安全多方学习所使用的底层原语以及隐私保护技术，将其分成了四种，即基于同态加密的推理方案，基于混淆电路的推理方案，基于混合协议推理的方案和基于秘密共享的训练方案。此外，我们分析了 19 个主流的安全多方学习框架。



图表 3 隐私机器学习框架分类图

4.1 基于同态加密的方案

Nathan Dowlin 等人[12]首次提出的 CryptNets 框架将同态加密与机器学习结合，为安全多方学习提供了创新性的解决方案。在该框架中，数据所有者使用同态加密对数据进行加密，然后将其发送至云服务器。云服务器在这些加密数据上执行神经网络模型的安全预测。采用了加密算术库（如 SEAL）进行同态加密。在针对 MNIST 数据集的实验中，CryptNets 在单个 PC 上实现了每小时 58982 个预测的吞吐量，延迟为 250 秒，准确率达到 99%。

在 CryptNets 基础上，BNormCrypt[5]进行了改进，采用低阶多项式逼近非线性函数，并引入了批量归一化，从而完善了 CNN 网络。CryptoDL[16]在 CryptNets 的基础上进一步改进，理论上证明在一定误差范围内能够找到函数的最低次多项式近似。该框架设计了逼近 CNN 中常用激活函数（如 ReLU、Sigmoid 和 Tanh）的方法，相较于 BNormCrypt 更准确，这对于有效的同态加密方案至关重要。此外，CryptoDL 通过使用平均池化替代最大池化层，提高了整体框架的运行效率。

尽管基于同态加密的框架在通信开销和通信轮次方面具有较低的优势，但通常伴随着巨大的计算复杂度和较高的内存占用。虽然同态加密适用于计算线性函数，如矩阵乘法，但在处理非线性函数（如 ReLU 和 Sigmoid）时效率较低。此外，由于密文爆炸的问题，基于同态加密的安全多方学习框架尚不适用于深度神经网络的安全训练，通常仅用于机器学习模型的安全预测场景。在这些场景中，数据持有者使用同态加密加密数据，并将其发送至服务器，服务器进行模型预测后返回加密的结果，最后由数据所有者解密以完成机器学习模型的安全预测。

4.2 基于混淆电路的方案

Rouhani 等人[32]提出了 DeepSecure 框架，是基于混淆电路的安全两方深度学习预测方案。考虑到半诚实的安全模型，DeepSecure 通过对数据和网络结构进行预处理，在安全计算开始之前加速在线阶段的运算速度。这是首个能够对分布式客户端生成的数据进行准确可扩展的深度学习分析，同时不牺牲安全性以维持

效率的框架。DeepSecure 解决了同态加密非线性计算复杂、效率低下的问题，采用了细粒度和粗粒度数据以及深度网络并行性，以避免 Yao 的 GC 协议执行过程中不必要的计算和通信。XONN[30]是在 DeepSecure 基础上引入二进制神经网络（BNN）的改进版本，提供了编译器，将模型描述从高级 Python（如 Keras）转换为 XONN 的编译器。此外，XONN 还提出了基于不经意传输（OT）的条件加法协议，该协议优化了网络输入层的昂贵计算，比传统的 GC 协议快 6 倍。

基于混淆电路的框架主要用于两方场景，通常具有常数通信轮次，但通信开销与电路大小成正比，因此通信量较大。由于混淆电路是按比特进行计算的，对于计算线性操作（如矩阵乘法），计算复杂度和通信开销较大，但在计算非线性函数（如比较）时较为高效。因此，基于混淆电路的安全多方学习框架通常适用于简单的机器学习模型训练，如逻辑回归。对于复杂模型，如神经网络，这些框架通常仅适用于安全预测任务。

4.3 基于混合协议推理的方案

前述的单一协议方案各有其局限性，因此有效结合这些方案的长处，克服各自的弱点，成为提高效率的一种可能途径。混合协议的基本思想是将同态加密和混淆电路两者结合，以便在进行安全计算时可以有效地利用它们的长处。同态加密适用于表示为算术电路的操作，如加法和乘法，而混淆电路适用于表示为布尔电路的操作，如比较。然而，协议之间的转换通常是昂贵的。以下将详细介绍混合协议的推理方案，而基于秘密共享的方案将在后文中进行详细阐述。

MiniONN[24]是首个能够将任何常见神经网络模型转换为无关神经网络的技术，而无需修改训练阶段。该方案使用同态加密和秘密共享技术处理线性层，而对于非线性层则采用混淆电路，巧妙地规避了单一协议的弱点。GAZELLE[20]在 MiniONN 的基础上提供了一个可扩展的低延迟安全神经网络推理系统，混合使用同态加密和混淆电路。线性层使用同态加密的方法，非线性层则采用混淆电路。该方案设计了一个同态加密库，支持将多个纯文本打包到单个密文中，提供了 SIMD（单指令多数据）同态加法（SIMDAdd）和标量乘法（SIMDScMult）以及置换明文时隙（Perm）等操作。GALA[40]对 GAZELLE 进行了改进，将同态加密的线性计算视为一系列同态 Add、Mult 和 Perm 操作，通过选择每个线性计算步骤中最昂贵的操作来降低总体成本。该方案引入了逐行权重矩阵编码和 Add-second-Perm 方法（内核分组）来减少卷积的 Perm 操作。在 2023 年，Xuanang Yang 等人[37]对 GALA 进行了优化，以同态方式进行矩阵向量积和卷积的计算方法，从而显著减少通信轮数，并避免了昂贵的置换操作，使矩阵向量积和卷积方法比 GALA 更快 $2.9-3.4$ 和 $2.9-4$ 倍。

Mishra 等人[25]在 GAZELLE 的基础上提出了 Delphi，通过将同态计算从在线阶段移至离线阶段来降低在线服务时延。对于非线性层，Delphi 使用多项式近似激活函数的安全计算友好的部分代替 ReLU 函数，利用神经网络架构搜索和超参数优化技术自动化 ReLU 函数替换过程，从而显著提高了隐私推理的效率。Circa[10]继承了 Delphi 的离线在线模式，主要对非线性计算进行了优化，将 ReLU

重新表述为近似符号测试，并引入一种新的截断方法，极大地降低了每个 ReLU 的成本。Gforce[28]设计了随机舍入和截断 (SRT) 层，使量化感知训练方案 SWALP 更符合加密工具。SWALP 在低精度设置下训练 DNN，同时保持准确性。为线性层和公共非线性层提出了一套 GPU 友好的协议。

在基于混合协议的框架中，不同的协议相互补充，充分发挥各自的优势。混合协议的应用广泛，特别适用于机器学习模型的安全预测。然而，不同协议之间的转换仍然是昂贵的，需要在混合协议带来的性能提升与协议转换的开销之间进行权衡。混合协议的组合方式包括同态加密+混淆电路，秘密共享+混淆电路，以及同态加密+混淆电路+秘密共享等。

4.4 基于秘密共享的解决方案

ABY[7]提出了一种创新的混合协议框架，将算术共享、布尔共享和姚氏混淆电路等不同的安全多方计算路线有效结合。ABY 框架在安全双方计算中引入了最佳实践，通过 A、B、Y 之间的转换，促进了混合协议的发展，为隐私机器学习奠定了基础。SecureML[27]基于 ABY 构建，是首个半诚实隐私保护机器学习系统，支持两方服务器模型。该系统提供了隐私保护的线性回归、逻辑回归和神经网络训练协议。ABY2.0[29]对 ABY 协议进行了改进，实现了更高效的半诚实安全两方计算，特别关注了在线阶段的性能提升。该版本还提供了高效的协议支持，涵盖了标量积、矩阵乘法、比较、最大池和等价性测试等基本原语。

Chameleon[31]引入了一种新型混合协议框架，专注于安全函数评估（SFE）。该框架允许双方在不公开私有输入的情况下协同计算函数，采用了伪 3PC 的设计，实质上是基于两方计算（2PC），通过引入可信第三方辅助计算，显著提高了生成乘法三元组的效率。

ABY3[26]在 ABY 的基础上进行了拓展，将混合协议扩展到了三方，并对秘密共享进行了优化。该框架实现了三方下的算术共享、布尔共享和姚氏混淆电路的转换协议。为适应医疗数据计算，本文在三方 ABY 协议基础上进行了扩展和细化，以提高其在医疗数据处理中的适用性。

SecureNN[34]是专为隐私机器学习优化的框架，提供了新颖的三方安全计算协议，支持矩阵乘法、卷积、纠正线性单位、Maxpool、归一化等神经网络构建。SecureNN 通过消除昂贵的混淆电路和忽略的传输协议来显著改善通信效率，采用数值计算的近似方式构建非线性函数，为各种深度学习网络提供了更好的支持。SecureNN 还是第一个在卷积神经网络（CNN）上提供神经网络训练的框架。在本文中，我们基于 ABY 框架参考 SecureNN 的思路，将三方 ABY 扩展到支持 CNN。

Falcon[35]提出了一种端到端的第三方协议，可有效进行大型机器学习模型的私人训练和推理。该协议采用了 ABY3 的算术共享，但舍弃了混淆电路，转而采用 SecureNN 的方法构建非线性计算。这使得 Falcon 协议在高效计算的情况下可以支持高容量网络，例如 VGG16。然而，对于大型模型，计算时间相应增加，例如在 VGG16 上需要 14 天的时间。

Piranha[36]是一个通用、模块化的平台，用于通过 GPU 加速基于秘密共享的多方计算协议。Piranha 提供了当前通用 GPU 库中不存在的基于整数的内核，并实现了模块化协议支持。该平台允许应用程序完全不了解其使用的底层协议，通过 Piranha 进行 GPU 加速，Falcon 在训练 VGG16 上的时间从 14 天缩短到 1 天多。

基于秘密共享的安全多方学习框架通常计算复杂度较低，通信量和通信轮次与电路深度成正比。这些框架在计算线性操作时通常具有高效性，但在处理非线性操作时，例如比较，其效率较低。这些框架可用于机器学习模型的安全预测和训练，适用于安全外包计算场景，将数据持有方和参与方分离，支持任意数量的数据持有方。在基于秘密共享的安全多方学习中，各参与方进行本地计算，根据使用的秘密共享协议，适用于不同的安全模型场景。

4.5 方案对比

安全多方学习框架在设计时确实需要综合考虑各种因素，包括通信开销、计算复杂度、内存占用等，以在实际应用中取得平衡。

- 同态加密技术本身具有计算复杂度高、密钥及密文膨胀、高内存占用等缺点。然而，其在两个参与方只交互一次的情况下完成学习任务，更适合于安全预测的功能场景。
- 混淆电路技术通信轮次恒定，适合计算非线性函数。然而，通常具有巨大的通信开销，更适合与其他技术混合使用，特别是用于安全预测的功能场景。
- 基于混合协议的框架充分利用了不同协议的优势，特别是混淆电路用于非线性函数计算。然而，协议之间的转换可能引入昂贵的开销，需要在性能提升和协议转换开销之间进行权衡。这种框架广泛应用于机器学习模型的安全预测。
- 基于秘密共享的安全多方学习框架通常具有较低的计算复杂度，然而，在计算非线性操作时效率相对较低。其通信开销和计算复杂度与电路深度成正比，适用于各种场景，支持安全训练和安全预测。

目前大部分安全多方学习框架只能支持诚实大多数/不诚实大多数的半诚实安全模型，以及诚实大多数的恶意安全模型，而不诚实大多数的恶意安全模型只有基于 SPDZ 协议的安全多方学习框架才可以支持。

5 最新研究进展

5.1 模型压缩加速

模型压缩（Model Compression）旨在通过各项技术手段降低机器学习模型的大小、复杂度和计算量，以便在资源受限的设备上进行高效部署和运行。这一领域的研究旨在有效减少模型的存储和计算资源需求，提高推理速度和效率，从而在资源受限的环境中实现机器学习应用的高效性。结合模型压缩和隐私机器学习，可以有效解决密码学协议效率问题。主要的模型压缩技术包括模型蒸馏、量化、

剪枝和参数共享。

- 模型蒸馏(Distillation)，使用大模型的学到的知识训练小模型，从而让小模型具有大模型的泛化能力。
- 量化(Quantization)，降低大模型的精度，减小模型。
- 剪枝(Pruning)，去掉模型中作用比较小的连接。
- 参数共享，共享网络中部分参数，降低模型参数数量。

现在已有的隐私机器学习模型加速方案有 NASS[1]，CryptoNAS[11]和 DeepReDuce[18]。NASS 是一个集成框架，专门用于搜索推理设计的定制网络架构。它将加密协议建模为设计元素，通过联合优化中的预测超参数，找到平衡预测准确性和执行效率的最佳网络架构。CryptoNAS 根据 PI 的需求定制模型，保持线性层不变，减少非线性。它提出了对 ReLU 高效网络的优化，采用了 ReLU 修剪和混洗来降低计算成本，同时保持模型平衡。DeepReDuce 采用了删除经典网络中的 ReLU 的策略，显著减少推理延迟并保持高精度。该方法通过输出一个网络帕累托边界，权衡 ReLU 数量和准确性，将准确率提高到 3.5\%（iso-ReLU 计数），同时在保持准确性的前提下将 ReLU 减少了 3.5 倍（iso-accuracy）。

5.2 隐私大模型推理

随着深度学习技术的不断发展，大型模型推理逐渐成为学术界和产业界共同关注的焦点。GPT 展现出卓越的语言理解和生成能力，以及广泛的知识覆盖，因而在业内外获得一致好评。然而，这也暴露了一系列安全隐患，隐私机器学习计算被提出以增强其安全性。近期，大量学者开始研究隐私保护大型模型推理。

MPCFormer[22]主要优化大型模型中的非线性部分，通过对 GeLU 和 Softmax 函数进行 MPC 友好的近似。通过知识蒸馏，该方法提高了后续模型的准确率。另一方面，MPCViT[39]提出了在 Transformer 中不同 Attention 和 GeLU 对模型性能影响不同的观点，以更细粒度的方式实现模型简化。MPCViT 研究了多种 Attention 的各个指标，选择了基于 Scaling 和 ReLU 的 Attention 近似方案，通过细粒度地替换模型中的 Attention，并尝试将 GeLU 线性化，从而合并 FFN 中的两个全连接层。通过模型压缩加速方法，对简化后的模型进行再次训练以提升性能。

Iron[14]在 SIRNN、Cheetah 和 CryptFlow2 的基础上，针对 Transformer 中的各个算子提出了具体的两方计算协议实现，成功实现了两方下的模型推理，但未实现安全 Embedding。相比之下，PUMA[8]在 ABY3 的基础上，系统性地实现了 Transformer 的安全推理，包括安全 Embedding。该方案使用户能够对模型实现“零”修改即可直接运行安全预测。这主要得益于对 GeLU 函数和 exp 函数的高精度拟合。PUMA 是首个公开实现 LLaMA-7B 级别模型安全推理的方案。CipherGPT[17]针对 GPT 模型提出了基于 subfield-VOLE 技术的预处理打包优化方法，通过不平衡矩阵乘法大大减少矩阵乘法的预处理开销。对于 GeLU，采用了分段拟合技术，并使用 SIRNN 中的多项技术来优化近似多项式的安全计算。CipherGPT 首次对 GPT 的输出层进行了详细的协议设计，构建了安全 Top-K 和

采样协议，更符合明文下 GPT 的 vec2word。

SIGMA[13]基于函数秘密分享技术（FSS），在 2+1-Party（2 Computing Servers + 1 Dealer）场景下重新构造了 Transformer 的各个函数的安全计算协议，并提供了 CPU 和 GPU 的两个具体版本。该方法显著提升了安全预测的效率。这些研究共同为隐私保护大型模型推理领域的发展做出了重要贡献。

6 讨论与发展建议

本章讨论了安全多方学习与其他隐私保护机器学习技术，如联邦学习的区别，并提出了安全多方学习未来的发展建议。

6.1 现阶段技术对比

现阶段隐私机器学习的主要技术是采用联邦学习。但基于联邦学习的隐私计算任然存在很多安全隐患，无法做到真正的安全。在基于联邦学习的隐私计算系统中，通常需要考虑到以下关键问题以确保系统的有效性和安全性：

- 构建可信的中央服务器：中央服务器在 FL 中扮演着重要的角色，用于聚合局部模型梯度以建立全局模型。在保护用户隐私的前提下，构建一个可信的服务器至关重要。
- 确保可靠的客户端-服务器通信：客户端和服务器之间的通信必须是安全和可靠的。攻击者可以在客户端和服务器之间建立的通信通道上部署数据攻击，以窃取更新的信息，从而中断全局服务器处的模型计算。
- 处理非独立同分布问题：数据集的非独立同分布问题可能导致数据训练的高度分散性。为了解决这个问题，需要开发多种解决方案，例如创建额外的数据子集，以在客户端之间进行公平分配，从而确保在基于 FL 的智能医疗中进行有效的数据训练。
- 安全参数设置：在 FL 中，分布式数据上传梯度的安全性是通过采用同态加密和差分隐私来保护的。然而，同态加密在大数计算下的运算效率较低，并且差分隐私的应用可能导致模型精度下降。因此，需要恰当地设置参数以在安全性和精度之间取得平衡。

尽管联邦学习可以解决单一的数据问题，但无法彻底解决数据隐私问题，第三方可信的服务器在实践操作中，存在巨大的安全隐患。此外，FL 系统中还存在其他潜在的安全攻击，如中毒攻击、推理攻击、后门攻击、恶意服务器、通信瓶颈和搭便车攻击等。这些攻击的不断出现，让联邦学习变的不那么可靠。为了提高安全性，安全多方计算技术的研究不可或缺。

6.2 未来发展建议

安全多方计算作为目前数据隐私计算的一个方法，虽然已经取得了一些研究成果，但是在具体应用环境中，其计算效率仍然需要进一步优化。本文广泛调研了同领域其他安全多方计算框架，其普遍存在本文实验暴露的精度损失的问题。当然，隐私计算作为一个新的领域，安全多方计算前尚处在起步阶段。本文建议可以从以下几个方面展开研究:

首先，想要大规模部署 MPC，标准化是一个必要的步骤。然而，这不是一项简单的任务，因为存在许多在安全性和效率方面具有不同优势的不同类型的 MPC 协议。此外，MPC 设计中使用了许多技术和不同的假设。这些使得 MPC 标准化过程变得很困难。

其次，针对 MPC 的设计过程中，需要更加高效的协议和方法研究，这不仅需要算法层面的突破，也需要要专门的硬件优化，比如将 MPC 搭载到 GPU 进行加速训练。这将极大提高 MPC 的计算效率，在提升效率的同时，也应该注重多样化发展，以支持更多的机器学习算法和模块，如 XGboost，Transformer 等。

最后，MPC 的安全性是极其重要的，对于恶意设置中的诚实多数 MPC，最近的一些工作[2,21,6]设计的 MPC 协议可以支持大规模的恶意安全。然而，它们的具体效率仍然不高。构建具有更高具体效率的大规模恶意安全 MPC 协议，并为数千方提供有效的实现扩展将是一个有趣的未来工作。

参考文献

[1] S. Bian, W. Jiang, Q. Lu, Y. Shi, and T. Sato. Nass: Optimizing secure inference via neural architecture search. arXiv preprint arXiv:2001.11854, 2020.

[2] M. Byali, H. Chaudhari, A. Patra, and A. Suresh. Flash: fast and robust framework for privacy-preserving machinelearning. Cryptology ePrint Archive, 2019.

[3] J. Cabrero-Holgueras and S. Pastrana. Sok: Privacy-preserving computation techniques for deep learning. Proc. Priv. Enhancing Technol., 2021(4):139 – 162, 2021.

[4] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In Proceedings 42nd IEEESymposium on Foundations of Computer Science, pages 136 – 145. IEEE, 2001.

[5] H. Chabanne, A. De Wargny, J. Milgram, C. Morel, and E. Prouff. Privacy-preserving classification on deep neuralnetwork. Cryptology ePrint Archive, 2017.

[6] H. Chaudhari, R. Rachuri, and A. Suresh. Trident: Efficient 4pc framework for privacy preserving machine learning. arXiv preprint arXiv:1912.02631, 2019.

[7] D. Demmler, T. Schneider, and M. Zohner. Aby-a framework for efficient mixed-protocol secure two-party computation. In NDSS, 2015.

[8] Y. Dong, W.-j. Lu, Y. Zheng, H. Wu, D. Zhao, J. Tan, Z. Huang, C. Hong, T. Wei, and W. Cheng. Puma: Secure inference of llama-7b in five minutes. arXiv preprint arXiv:2307.12533, 2023.

[9] D. Evans, V. Kolesnikov, M. Rosulek, et al. A pragmatic introduction to secure multi-party computation. Foundations and Trends® in Privacy and Security, 2(2-3):70 – 246, 2018.

[10] Z. Ghodsi, N. K. Jha, B. Reagen, and S. Garg. Circa: Stochastic relus for private deep learning. Advances in Neural Information Processing Systems, 34:2241 – 2252, 2021.

[11] Z. Ghodsi, A. K. Veldanda, B. Reagen, and S. Garg. Cryptonas: Private inference on a relu budget. Advances in Neural Information Processing Systems, 33:16961 – 16971, 2020.

[12] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In M. F. Balcan and K. Q. Weinberger, editors, Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, pages 201 – 210,

New York, New York, USA, 20 – 22 Jun 2016. PMLR.
[13] K. Gupta, N. Jawalkar, A. Mukherjee, N. Chandran, D. Gupta, A. Panwar, and R. Sharma. Sigma: Secure gpt inference with function secret sharing. Cryptology ePrint Archive, 2023.
[14] M. Hao, H. Li, H. Chen, P. Xing, G. Xu, and T. Zhang. Iron: Private inference on transformers. Advances in Neural Information Processing Systems, 35:15718 – 15731, 2022.
[15] M. Hastings, B. Hemenway, D. Noble, and S. Zdancewic. Sok: General purpose compilers for secure multi-party computation. In 2019 IEEE symposium on security and privacy (SP), pages 1220 – 1237. IEEE, 2019.
[16] E. Hesamifard, H. Takabi, M. Ghasemi, and R. N. Wright. Privacy-preserving machine learning as a service. Proc. Priv. Enhancing Technol., 2018(3):123 – 142, 2018.
[17] X. Hou, J. Liu, J. Li, Y. Li, W.-j. Lu, C. Hong, and K. Ren. Ciphergpt: Secure two-party gpt inference. Cryptology ePrint Archive, 2023.
[18] N. K. Jha, Z. Ghodsi, S. Garg, and B. Reagen. Deepreduce: Relu reduction for fast private inference. In International Conference on Machine Learning, pages 4839 – 4849. PMLR, 2021.
[19] H. JIANG, Y. LIU, X. SONG, H. WANG, Z. ZHENG, and Q. XU. Cryptographic approaches for privacy-preservingmachine learning. 电子与信息学报, 42(5):1068 – 1078, 2020.
[20] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. Gazelle: A low latency framework for secure neural network inference. In 27th USENIX Security Symposium (USENIX Security 18), pages 1651–1669, 2018.
[21] N. Koti, M. Pancholi, A. Patra, and A. Suresh. Swift: Super-fast and robust privacy-preserving machine learning. In USENIX Security Symposium, pages 2651–2668, 2021.
[22] D. Li, R. Shao, H. Wang, H. Guo, E. P. Xing, and H. Zhang. Mpcformer: fast, performant and private transformer inference with mpc. arXiv preprint arXiv:2211.01452, 2022.
[23] Y. Lindell. Secure multiparty computation (mpc). Cryptology ePrint Archive, 2020.
[24] J. Liu, M. Juuti, Y. Lu, and N. Asokan. Oblivious neural network predictions via minionn transformations. In Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pages 619–631, 2017.
[25] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa. Delphi: a cryptographic inference system for neural networks. In Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, pages 27–30, 2020.
[26] P. Mohassel and P. Rindal. Aby3: A mixed protocol framework for machine learning. In Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pages 35–52, 2018.
[27] P. Mohassel and Y. Zhang. Secureml: A system for scalable privacy-preserving machine learning. In 2017 IEEE symposium on security and privacy (SP), pages 19–38. IEEE, 2017.
[28] L. K. Ng and S. S. Chow. {GForce}:{GPU-Friendly} oblivious and rapid neural network inference. In 30th USENIX Security Symposium (USENIX Security 21), pages 2147–2164, 2021.
[29] A. Patra, T. Schneider, A. Suresh, and H. Yalame. Aby2. 0: Improved mixed-protocol secure two-party computation. In USENIX Security Symposium, pages 2165–2182, 2021.
[30] M. S. Riazi, M. Samragh, H. Chen, K. Laine, K. Lauter, and F. Koushanfar. {XONN}:{XNOR-based} oblivious deep neural network inference. In 28th USENIX Security Symposium (USENIX Security 19), pages 1501–1518, 2019.
[31] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar.

Chameleon: A hybridsecure computation framework for machine learning applications. In Proceedings of the 2018 on Asia conference on computer and communications security, pages 707–721, 2018.

[32] B. D. Rouhani, M. S. Riazi, and F. Koushanfar. Deepsecure: Scalable provably-secure deep learning. In Proceedingsof the 55th annual design automation conference, pages 1–6, 2018.

[33] L. Song, G. Lin, J. Wang, H. Wu, W. Ruan, and W. Han. Sok: Training machine learning models over multiple sources with privacy preservation. arXiv preprint arXiv:2012.03386, 2020.

[34] S. Wagh, D. Gupta, and N. Chandran. Securenn: 3-party secure computation for neural network training. Proc. Priv. Enhancing Technol., 2019(3):26–49, 2019.

[35] S. Wagh, S. Tople, F. Benhamouda, E. Kushilevitz, P. Mittal, and T. Rabin. F: Honest-majority maliciously secure framework for private deep learning. Proceedings on Privacy Enhancing Technologies, 2021(1):188–208, 2021.

[36] J.-L. Watson, S. Wagh, and R. A. Popa. Piranha: A gpu platform for secure computation. In 31st USENIX Security Symposium (USENIX Security 22), pages 827–844, 2022.

[37] X. Yang, J. Chen, K. He, H. Bai, C. Wu, and R. Du. Efficient privacy-preserving inference outsourcing for convolutional neural networks. IEEE Transactions on Information Forensics and Security, 2023.

[38] A. C. Yao. Protocols for secure computations. In 23rd annual symposium on foundations of computer science (sfcs1982), pages 160–164. IEEE, 1982.

[39] W. Zeng, M. Li, W. Xiong, T. Tong, W.-j. Lu, J. Tan, R. Wang, and R. Huang. Mpcvit: Searching for accurate and efficient mpc-friendly vision transformer with heterogeneous attention. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 5052–5063, 2023.

[40] Q. Zhang, C. Xin, and H. Wu. Gala: Greedy computation for linear algebra in privacy-preserved neural networks. arXiv preprint arXiv:2105.01827, 2021.

[41] 何玉长，王伟. 数据要素市场化的理论阐释. 当代经济研究, 308(4):33‐44, 2021.

[42] 贺小石. 大数据背景下公民信息安全保障体系构建——兼论隐私政策的规制原理及其本土化议题. 中国特色社会主义研究,3(6):100–109, 2022.

[43] 高莹, 李寒雨, 王玮, 刘翔, 陈洁. 不经意传输协议研究综述. 软件学报, pages 12‐19, 2022.