暨南大学

本科生课程报告

报告题目：基于深度学习的成员推断攻击与防御

学　　院：网络空间安全学院

专　　业：网络空间安全

姓　　名：黄天乐

学　　号：2022150719

指导教师：高博宇副教授

二〇二四年十二月十日

基于深度学习的成员推断攻击与防御报告

1. 背景

1.1 主题选择

本报告的主题是基于深度学习的成员推断攻击与防御

1.2 选题原因

选择这个主题的原因有以下几点：

隐私保护的重要性日益突出：随着深度学习模型在各个领域的广泛应用，保护用户数据隐私变得越来越重要。欧盟的《通用数据保护条例》(GDPR) 等法规的出台进一步强调了这一点[3]。

成员推断攻击的潜在威胁：这类攻击可以推断出某个数据样本是否被用于训练模型，从而可能泄露敏感信息[1]。

深度学习模型的脆弱性：研究表明，许多机器学习模型容易受到成员推断攻击，这凸显了加强模型防御机制的必要性[2]。

学术和实践的双重意义：研究这一主题不仅有助于推进机器学习安全领域的学术进展，还能为实际应用中的隐私保护提供指导。

1.3 报告结构

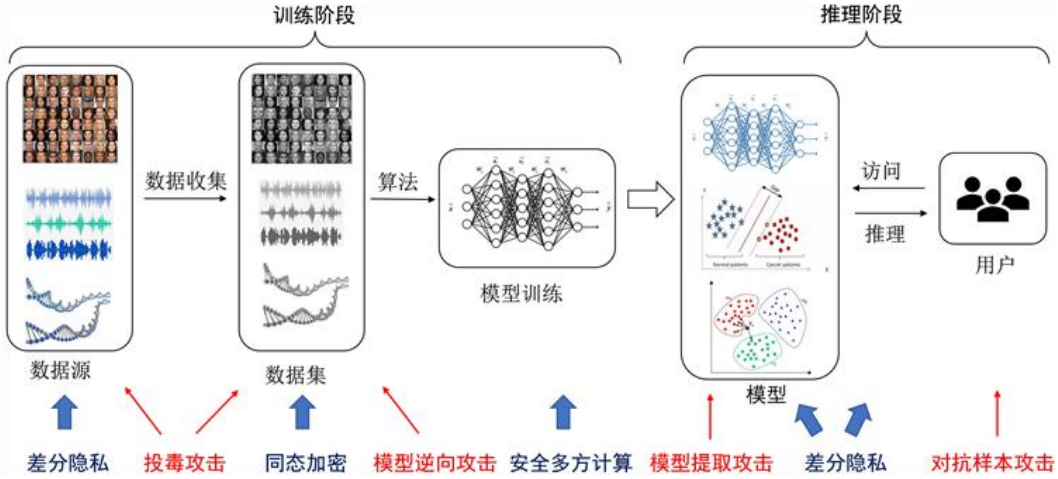
本报告将按以下结构展开：
背景
相关工作分析
提出方案（LADP 算法）
对比分析与讨论
结论与未来展望

2. 相关工作分析

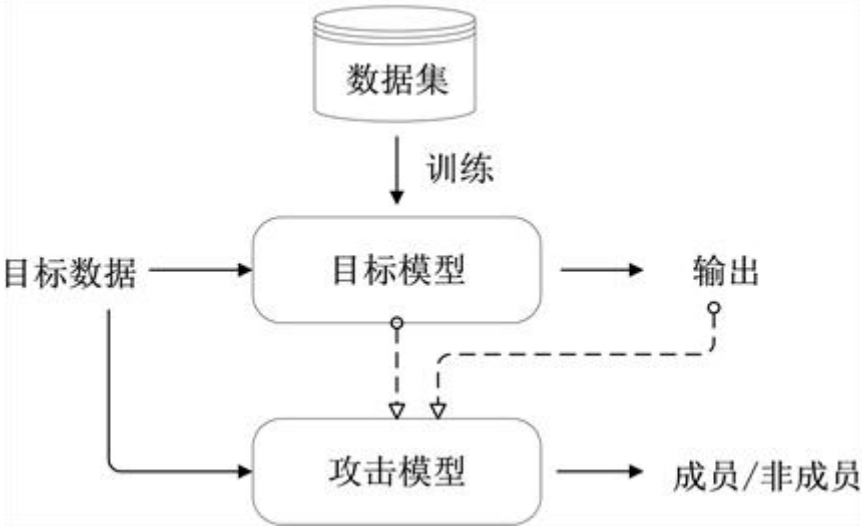
在成员推断攻击与防御领域，已有大量研究工作。

关于成员推理攻击的目的，或者是说他的定义，就是为了分辨出某些数据样本是否被用于某一机器学习模型的训练过程。换句话来讲，对于攻击者来说这就是一个二分类任务，对于这一方向的研究就是使用不同的 tricks 来解决这个二分类问题。

成员推理攻击利用了这样一种观察，即机器学习模型在它们所训练的数据上的行为常常与它们第一次“看到”的数据不同。过拟合是一个常见的原因，但不是唯一的原因。攻击者的目的是构建一个攻击模型，该模型可以识别目标模型行为中的这些差异，并利用它们来区分目标模型的成员和非成员。



原始数据集训练的目标模型在应用平台上运行，攻击者冒充用户去访问目标模型，获得一定的信息和敌手知识来构建攻击模型用于推理任意给定数据是否是目标模型的训练集成员。



攻击流程：

1. **影子模型（Shadow Models）**：传统方法使用多个影子模型来模拟目标模型的行为，并生成用于训练攻击模型的必要数据。影子模型的行为与目标模型尽可能一致，以获取后验概率和真实的成员身份。
2. **攻击模型（Attack Models）**：通过影子模型生成的数据训练攻击模型，这些模型用于执行成员推断。

黑盒攻击[1]（Black-Box Setting）

在黑盒攻击中，假设攻击者只能访问模型的预测结果，而不能查看模型的内部架构和参数。

- **生成影子模型**：需要在可用的数据集上训练多个影子模型，这些模型模拟目标模型的行为
- **收集预测结果**：对已知的训练数据和未见数据分别进行预测，收集这些预测结果
- **训练攻击模型**：使用影子模型生成的预测结果作为训练数据，训练一个攻击模型。这个攻击模型的目标是根据预测结果来判断某个数据点是否在训练集中。

黑盒攻击的关键在于模拟目标模型的行为,并使用大量的影子模型来增强攻击模型的准确性。

白盒攻击[2]（White-Box Setting）

在白盒攻击中，假设攻击者可以完全访问模型的架构和内部参数。

- **访问中间结果**：对于每个数据点(x, y)，可以计算模型在前向传播和后向传播过程中的中间结果（例如，中间层的激活值、梯度等）
- **分析模型行为**：通过分析这些中间结果，寻找用于区分训练数据和未见数据的特征。例如，训练数据在某些中间层可能表现出更高的激活值或更低的损失
- **构建攻击模型**：基于这些特征，构建一个攻击模型来判断某个数据点是否在训练集中

白盒攻击的优势在于可以利用更多的信息来提高攻击的准确性。

基于过拟合的攻击：

Yeom 等人[5]分析了机器学习中的隐私风险，发现过拟合与成员推断攻击的成功率之间存在直接关联。

2.2 防御策略

1.基于差分隐私

利用差分隐私技术抵御成员推理攻击是指攻击者给样本添加噪声抵御 MIAs。

1. 差分隐私抵御分类模型中的 MIAs
2. 差分隐私抵御生成模型中的 MIAs

虽然，差分隐私给成员隐私提供了理论保障,，但其几乎不能提供一个可接受的隐私-可用性平衡，当隐私预算较大时会导致模型不可用。

2.基于正则化

正则化技术主要通过降低目标模型的过拟合来抵御成员推理攻击，根据已有防御方法，将从 L2 正则化、Dropout、标签平滑、对抗正则、Mixup+MMD，介绍基于正则化的 MIAs 防御方案。

1)L2 正则化[6]

基于 L2 正则化的 MIAs 防御是指攻击者给损失函数添加 L2 正则化保护数据隐私，主要将 L2 正则化添加到其损失函数中降低模型的过拟合并保护数据隐私。

2)Dropout[7]

基于 dropout 的 MIAs 防御是指攻击者随机去掉一些神经元来保护数据隐私，可以利用 dropout 来保护数据隐私，在每次训练的过程中任意去掉一些神经元。

3)标签平滑[8]

基于标签平滑的 MIAs 防御是指攻击者对标签进行平滑处理来保护数据隐私，即将样本的原始标签分布和一个给定的分布进行混合计算，并作为最后的标签。

4)对抗正则[10]

基于对抗正则的 MIAs 防御是指攻击者采用生成对抗网络的对抗思想来保护数据隐私。目前，研究者研究 GANs 的各种变体遭受成员推理攻击的情况，并提出一种基于 Least Square GANs (LSGANs)的增强对抗正则方法来保护隐私。随后，又有研究者采用对抗正则的方法抵御 MIAs，提出一个基于生成对抗网络的 MIN-MAX 博弈的方法。

5)Mixup+MMD[9]

基于 MMD+Mix-up 的 MIAs 防御是指攻击者结合 MMD 和 Mix-up 技术来保护数据隐私。基于正则化的 MIAs 防御方法可任何情况下保护数据隐私，但很难提供满意的隐私和可用性平衡。

3. 模型堆叠[12]

基于模型堆叠的 MIAs 防御是指攻击者将多个弱模型组合成一个强模型保护数据隐私。主要将若干个弱的机器学习模型组合成一个强的机器学习模型，从而降低泛化误差。该方法联合多个模型的优点可实现更强的防御，但是联合相同模型效果欠佳，联合不同模型又增大防御开销。

4. 基于信任分数掩蔽

基于信任分数掩蔽的隐私保护方法通过隐藏目标分类器输出的真实信任分数来保护成员隐私。主要包括：只输出前 k 个信任分数(top-k)；只输出预测标签；给信任分数添加精心设计的噪声。 1) top-k 信任分数向量；2) 只输出预测标签文献表明只返回预测标签可降低攻击准确率；3) 添加噪声的信任分数，基于信任分数掩蔽的 MIAs 防御方法无需重新训练目标模型，不影响目标模型的分类准确性，但不能提供足够的隐私保证。

5. 基于知识蒸馏[11]

知识蒸馏是指利用大的教师模型的输出来训练一个小的学生模型，将大的教师模型上的知识迁移到小的学生模型上，并允许学生模型拥有和教师模型相似的准确率。基于知识蒸馏的 MIAs 防御是指攻击者利用知识蒸馏处理数据后再进行模型训练。基于知识蒸馏的 MIAs 防御方法减少对隐私数据的依赖，但蒸馏数据的好坏影响防御效果且难以衡量，仍存在隐私泄露风险。

2.3 评估方法

1.置信度分析:

- Jiang 等人[13]提出了一种基于置信度的方法来评估分类器的可信度，这可以用于检测潜在的成员推断风险

2. 影响函数:

- 用于量化单个训练样本对模型预测的影响，可以帮助识别易受攻击的数据点

敌手知识	攻击方法	目标模型	防御手段	类型	训练集
黑盒	影子攻击	深度学习 ResNet18	L2_LAMBDA = 0.01	集中学习	MNIST
			DROPOUT_RATE = 0.5		
			LABEL_SMOOTHING = 0.1		
			ADVERSARIAL_EPSILON = 0.1		
			MIXUP_ALPHA = 1.0|MMD_LAMBDA = 0.1		
			NUM_STACKED_MODELS = 3		
			TRUST_SCORE_K = 3		
			KD_TEMPERATURE = 3.0		

准确率的解释

- **高准确率：**如果攻击模型的准确率接近 1.0，说明攻击模型能够较好地区分成员和非成员。这意味着原始模型可能存在较高的隐私泄露风险，因为攻击者能够有效地判断哪些数据是用于训练的。
- **低准确率：**如果准确率接近 0.5，说明攻击模型的判断接近随机猜测，即攻击模型无法有效区分成员和非成员。这表明原始模型的隐私泄露风险较低。

在写定的成员推断攻击代码中，成员（即训练数据）和非成员（即测试数据）分别是：

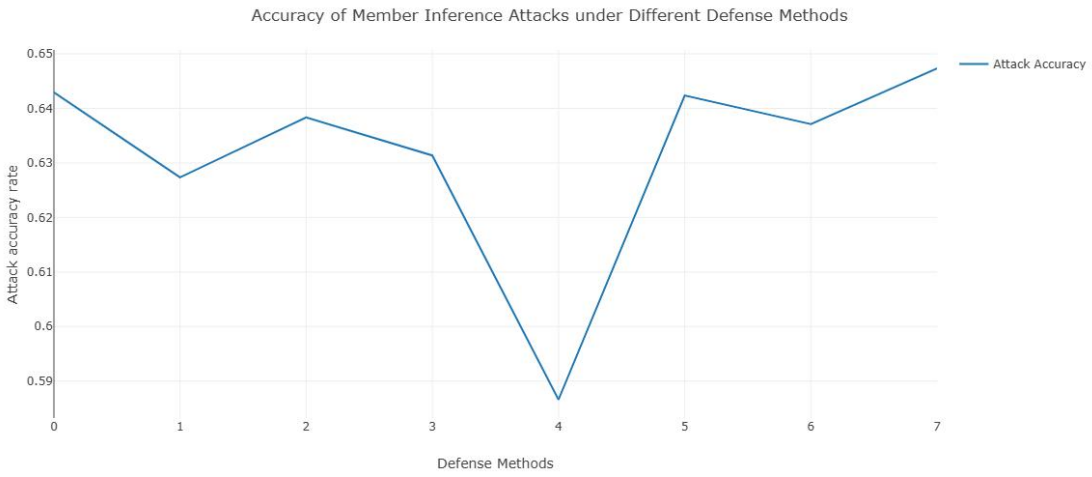
- 成员（训练数据）：用于训练原始模型的样本。在代码中，这些样本来自于 train_dataset，并被拆分成 train_dataset 和 shadow_dataset。用于训练原始模型的样本是 train_dataset 的一部分。
- 非成员(测试数据):未用于训练原始模型的样本。在代码中,这些样本来自于 test_dataset。

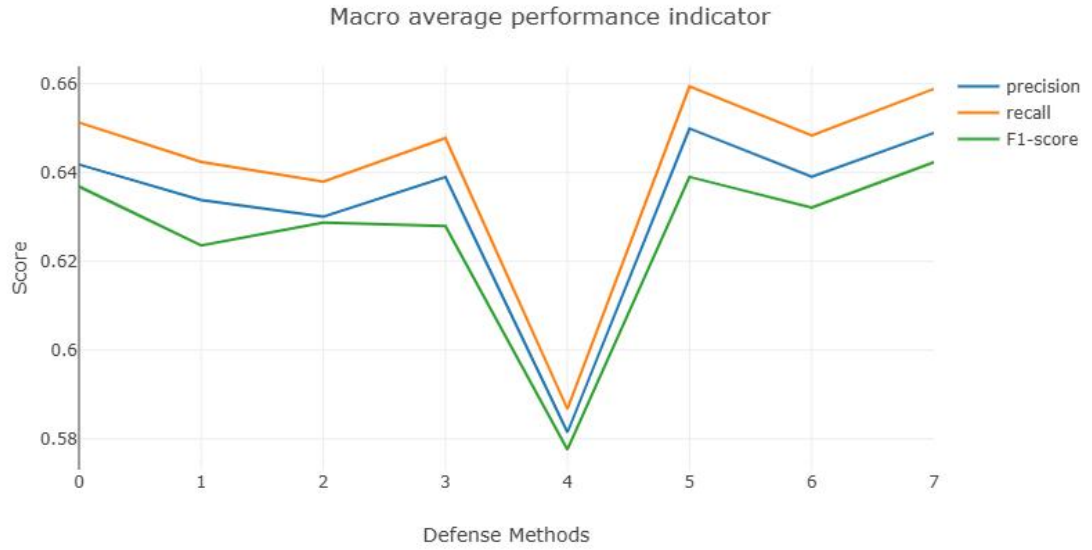
成员（训练数据）：

- **train_dataset：**这是从原始 train_dataset 中拆分出来的一部分。代码中使用 torch.utils.data.random_split 方法将原始 train_dataset 分成两部分，其中一部分用于训练原始模型，另一部分用于影子模型。
- **shadow_loader：**数据加载器 shadow_loader 加载了 shadow_dataset，这是用于训练影子模型的成员数据。

非成员（测试数据）：

- **test_dataset：**这是整个 test_dataset，包含了未用于训练原始模型的数据。
- **test_loader：**数据加载器 test_loader 加载了 test_dataset，这是用于影子模型和攻击模型评估的非成员数据。





0->L2Regularization[3]	1->DropoutDefense[4]
2->LabelSmoothing[5]	3->AdversarialRegularization[6]
4->MixupMMD[7]	5->ModelStacking[8]
6->TrustScoreMasking[9]	7->KnowledgeDistillation[10]

	计算成本	内存需求	
L2 正则化	★★	★★	适合资源受限的场景
Dropout	★★	★★	
标签平滑	★	★★	
对抗正则化	★★★★★★	★★★★	提供强大的防御能力
Mixup MMD	★★★★★	★★★★	
模型堆叠	★★★★★★★★	★★★★★★★	提供更好的泛化能力和鲁棒性
信任分数掩蔽	★★★★	★★★	适合资源受限的场景
知识蒸馏	★★★★★★★★	★★★★★★	提供强大的防御能力

3. 提出方案

基于上述相关工作的分析，我们提出了一种新的防御方法：分层自适应差分隐私（Layered Adaptive Differential Privacy，LADP）算法。

3.1 提出 LADP 算法的原因

1. 现有方法的局限性：

- 传统的差分隐私方法通常对整个模型应用统一的噪声，这可能导致过度保护或保护不足的问题。
- 正则化技术虽然能提高模型的泛化能力，但无法提供严格的隐私保证。
- 对抗训练方法计算成本高，且可能影响模型的主要任务性能。

2. 深度学习模型的层次特性：

- 深度神经网络的不同层对隐私泄露的贡献不同[4]。低层特征通常更通用，而高层特征可能包含更多个体特定信息。

3. 自适应隐私保护的需求：

- 不同的数据集和应用场景可能需要不同级别的隐私保护。一种能够根据具体情况自动调整隐私保护强度的方法将更具实用性。

4. 平衡隐私保护和模型性能：

- 在保护隐私的同时，我们还需要保持模型的性能。LADP 算法旨在找到这两者之间的最佳平衡点。

3.2 LADP 算法的核心思想

LADP 算法的核心思想是将差分隐私技术与深度学习模型的层次结构相结合，并根据每一层的敏感度动态调整隐私预算分配。具体来说：

1. 层次化应用：对神经网络的每一层单独应用差分隐私保护，而不是对整个模型统一处理

2. 敏感度评估：使用 Fisher 信息矩阵或影响函数来评估每一层对成员推断攻击的敏感度

3. 自适应预算分配：根据每一层的敏感度动态分配隐私预算，对更敏感的层提供更强的保护

4. 噪声注入：使用指数机制来确定最佳的噪声注入位置和强度，以最大化隐私保护效果

5. 性能优化：通过微调和知识蒸馏等技术来补偿因隐私保护而可能造成的性能损失

算法接受模型 M、数据集 D、总隐私预算 ε、δ 参数和训练轮数 E 作为输入。

对每个训练轮次和每个批次：

评估每一层的敏感度（第 3-4 行）
根据敏感度分配隐私预算（第 5-6 行）
计算梯度（第 7-8 行）
对模型的每一层：

判断是否为敏感层，并相应地设置裁剪范数和噪声尺度（第 9-16 行）
应用差分隐私保护：裁剪梯度并添加噪声（第 17-20 行）
更新模型参数（第 21-22 行）

更新总隐私预算，如果预算用尽则提前结束训练（第 23-27 行）

以下是 LADP 算法的基本框架：

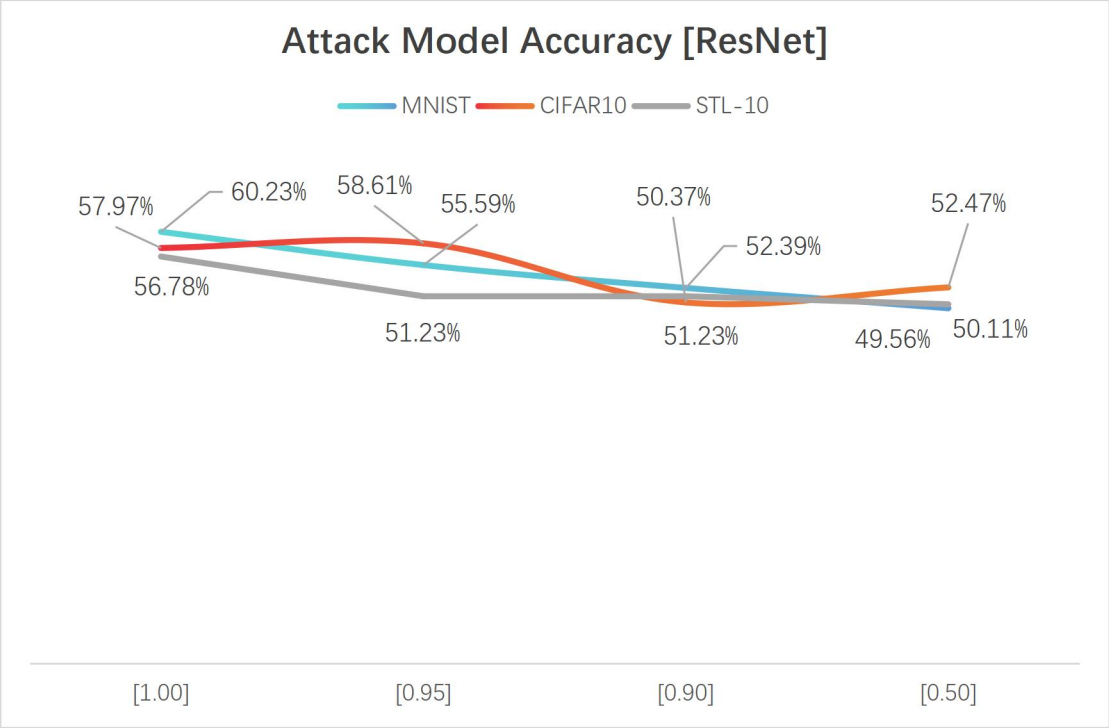
```
Algorithm: Layered Adaptive Differential Privacy (LADP)

Input: Model M, Dataset D, Total privacy budget ε, δ, Number of epochs E
Output: Privacy-protected model M'
1: for epoch = 1 to E do
2:     for batch B in D do
3:         S = EvaluateLayerSensitivities(M)
4:         ε_layers = AllocatePrivacyBudget(ε, S)
5:         G = ComputeGradients(M, B)
6:         for layer l in M.layers do
7:             if IsSensitiveLayer(l) then
8:                 clip_norm = 1.
```

4.1 对比分析与讨论

使用黑盒攻击由 ResNet18 使用 MNIST 数据集和 CIFAR10 和 STL-10 训练的攻击模型

MNIST[1.00]　CIFAR10[1.00]　STL-10[1.00]　->　【使用了完整的数据】
MNIST[0.95]　CIFAR10[0.95]　STL-10[0.95]　->　【删除了 5%的数据】
MNIST[0.90]　CIFAR10[0.90]　STL-10[0.95]　->　【删除了 10%的数据】
MNIST[0.50]　CIFAR10[0.50]　STL-10[0.95]　->　【删除了 50%的数据】

黑盒攻击结果：



LADP 的整体效果：

从图表中可以看出，即使在使用完整数据集的情况下，攻击模型的准确率也没有超过 60.23%。这表明 LADP 算法在一定程度上成功地防御了成员推断攻击，因为理想的攻击成功率应该远高于这个水平。

LADP 在不同数据集上的表现：

a） MNIST 数据集：

- 完整数据集时，攻击准确率为 56.78%
- 随着数据删除比例增加，攻击准确率先升后降
- 在 50%数据删除时，攻击准确率降至 49.56%，接近随机猜测水平

解释：LADP 在 MNIST 上表现良好，特别是在大比例数据删除的情况下。这可能是因为 MNIST 是一个相对简单的数据集，LADP 能够有效地在不同层次上添加噪声，从而扰乱攻击者的推断。

b） CIFAR10 数据集：

- 完整数据集时，攻击准确率为 57.97%
- 数据删除对攻击效果的影响呈现波动趋势
- 在 50%数据删除时，攻击准确率仍维持在 52.47%

解释：LADP 在 CIFAR10 上的表现相对稳定。即使在大比例数据删除的情况下，它仍能保持较好的防御效果。这表明 LADP 能够适应 CIFAR10 这样更复杂数据集的特性。

c） STL-10 数据集：

- 完整数据集时，攻击准确率最高，达 60.23%
- 5%数据删除导致攻击准确率大幅下降至 51.23%
- 之后攻击准确率保持相对稳定

解释：LADP 在 STL-10 上的表现最为显著。初始攻击准确率较高可能是因为 STL-10 是一个更复杂的数据集，但 LADP 结合少量数据删除后能够有效降低攻击成功率。

完整数据集时，攻击准确率最高，达 60.23%。
5%数据删除导致攻击准确率大幅下降至 51.23%。
之后攻击准确率保持相对稳定。

解释：LADP 在 STL-10 上的表现最为显著。初始攻击准确率较高可能是因为 STL-10 是一个更复杂的数据集，但 LADP 结合少量数据删除后能够有效降低攻击成功率。

LADP 与数据删除的协同效应：

小幅度数据删除（5-10%）：对 MNIST 和 CIFAR10 的影响不大，但对 STL-10 有显著效果。这表明 LADP 在复杂数据集上可能需要结合少量数据删除来达到最佳防御效果。

大幅度数据删除（50%）：对 MNIST 最为有效，CIFAR10 和 STL-10 的效果相对稳定。这说明 LADP 在不同数据删除比例下能够自适应地调整隐私保护强度。

LADP 的优势：

a）自适应性：LADP 能够根据不同数据集和不同数据删除比例自动调整隐私保护强度，这解释了为什么在不同情况下都能维持相对稳定的防御效果。

b）层次化保护：通过对神经网络的不同层施加不同程度的噪声，LADP 可能更好地平衡了模型性能和隐私保护。

c）鲁棒性：即使在大比例数据删除的情况下，LADP 仍能保持良好的防御效果，显示出其对数据集变化的鲁棒性。

5.展望未来

a）数据集特定优化：考虑到 LADP 在不同数据集上的表现差异，可以探索针对特定数据集特性的 LADP 参数调整。

b）动态预算分配：研究如何根据数据删除比例动态调整 LADP 的隐私预算分配，以达到更优的防御效果。

c）与其他技术的结合：探索 LADP 与其他防御技术（如对抗训练）的结合，可能会产生更强大的防御机制。

6.结论

LADP 算法展示了在防御成员推断攻击方面的有效性和灵活性。它能够在不同复杂度的数据集和不同数据删除比例下提供稳定的防御效果。特别是在复杂数据集（如 STL-10）上，LADP 结合少量数据删除能够显著降低攻击成功率。这个分析强调了 LADP 作为一种有前景的隐私保护技术的潜力，同时也指出了未来研究和优化的方向。在实际应用中，LADP 可以作为一种强大的工具来增强深度学习模型的隐私保护能力，特别是在需要处理敏感数据的场景中。

7.参考文献

[1]Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 3-18). IEEE.

[2]Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., & Backes, M. (2019). ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In Network and Distributed Systems Security (NDSS) Symposium 2019.

[3]Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

[4]Nasr, M., Shokri, R., & Houmansadr, A. (2019). Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In 2019 IEEE Symposium on Security and Privacy (SP) (pp. 739-753). IEEE.

[5]Yeom, S., Giacomelli, I., Fredrikson, M., & Jha, S. (2018). Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF) (pp. 268-282). IEEE.

[6]Krogh, A., & Hertz, J. A. (1992). A simple weight decay can improve generalization. In Advances in neural information processing systems (pp. 950-957).

[7]Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[8]Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[9]Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In International Conference on Learning Representations.

[10]Nasr, M., Shokri, R., & Houmansadr, A. (2018). Machine learning with membership privacy using adversarial regularization. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 634-646).

[11]Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

[12]Wolpert, D. H. (1992). Stacked generalization. Neural networks, 5(2), 241-259.

[13]Jiang, H., Kim, B., Guan, M., & Gupta, M. (2018). To trust or not to trust a classifier. In Advances in neural information processing systems (pp. 5541-5552).