深度学习数据窃取攻击在数据沙箱模式下的威胁分析与防御方法研究

潘鹤中[1]，韩培义[2]，向夏雨[1]，段少明[2]，庄荣飞[2]，刘川意[2,3]

（1. 北京邮电大学网络空间安全学院，北京 100876；2. 哈尔滨工业大学（深圳）计算机科学与技术学院，广东 深圳 518055；
3. 鹏城实验室网络空间安全中心，广东 深圳 518066）

摘 要：详细分析了数据沙箱模式下，深度学习数据窃取攻击的威胁模型，量化评估了数据处理阶段和模型训练阶段攻击的危害程度和鉴别特征。针对数据处理阶段的攻击，提出基于模型剪枝的数据泄露防御方法，在保证原模型可用性的前提下减少数据泄露量；针对模型训练阶段的攻击，提出基于模型参数分析的攻击检测方法，从而拦截恶意模型防止数据泄露。这 2 种防御方法不需要修改或加密数据，也不需要人工分析深度学习模型训练代码，能够更好地应用于数据沙箱模式下数据窃取防御。实验评估显示，基于模型剪枝的防御方法最高能够减少 73%的数据泄露，基于模型参数分析的检测方法能够有效识别 95%以上的攻击行为。

关键词：数据沙箱；数据窃取；AI 安全

中图分类号：TP309.2

文献标识码：A

DOI: 10.11959/j.issn.1000−436x.2021215

Threat analysis and defense methods of deep-learning-based data theft in data sandbox mode

PAN Hezhong[1], HAN Peiyi[2], XIANG Xiayu[1], DUAN Shaoming[2], ZHUANG Rongfei[2], LIU Chuanyi[2,3]

1. School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China
3. Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518066, China

Abstract: The threat model of deep-learning-based data theft in data sandbox model was analyzed in detail, and the degree of damage and distinguishing characteristics of this attack were quantitatively evaluated both in the data processing stage and the model training stage. Aiming at the attack in the data processing stage, a data leakage prevention method based on model pruning was proposed to reduce the amount of data leakage while ensuring the availability of the original model. Aiming at the attack in model training stage, an attack detection method based on model parameter analysis was proposed to intercept malicious models and prevent data leakage. These two methods do not need to modify or encrypt data, and do not need to manually analyze the training code of deep learning model, so they can be better applied to data theft defense in data sandbox mode. Experimental evaluation shows that the defense method based on model pruning can reduce 73% of data leakage, and the detection method based on model parameter analysis can effectively identify more than 95% of attacks.

Keywords: data sandbox, data theft, security of AI

收稿日期：2021−06−30；**修回日期**：2021−08−31

通信作者：刘川意，liuchuanyi@hit.edu.cn

基金项目：国家自然科学基金资助项目（No.61872110）

Foundation Item: The National Natural Science Foundation of China (No.61872110)

1 引言

人工智能（AI, artificial intelligence）时代，数据开放共享已然成为趋势，但是数据安全问题严重制约了大数据价值的发挥。数据沙箱模式，或称数据信托，是解决隐私保护和数据挖掘之间矛盾的有效方案[1-2]。数据沙箱分为调试环境和运行环境，数据拥有者将原始数据托管到运行环境中，并自动生成不包含隐私信息的样例数据。数据分析人员在调试环境中根据样例数据编写 AI 模型训练代码，并将其发送到运行环境。该代码在运行环境中对全量的原始数据进行分析，最终得到高可用性的 AI 模型，返还给数据分析人员。这一流程中，数据分析人员没有直接接触原始数据，又实现了 AI 模型在全量数据上的充分训练。

然而随着 AI 技术的发展，深度学习模型携带原始数据的能力逐渐增强。Carlini 等[3-4]从理论上证明了参数量庞大的语言模型会存储和泄露训练数据。基于这一观点，国内外专家学者对深度学习模型窃取训练数据的攻击手段展开了研究，文献[5-7]等都属于此类攻击。这类攻击可以作用于数据沙箱的运行环境，主动在深度学习模型中编码原始数据。例如 Song 等[7]提出可以通过在数据处理过程中生成恶意训练数据，或是在模型训练过程中引入恶意正则化项来窃取数据。这类攻击手段能够有效绕过数据沙箱对训练数据的隐私保护机制。目前，国内外研究当中尚缺乏对数据沙箱模式下深度学习数据窃取攻击的详细威胁分析和实验验证。

针对上述数据窃取攻击，国内外学者提出的防御手段主要分为两大类。一类是数据修改的方法，通过在原始数据中添加扰动或采用加密技术来保护训练数据[8-10]。这类方法虽然取得了一定进展但难以直接应用于数据沙箱模式中。其原因有两点：一是数据沙箱模式下，原始训练数据属于数据拥有者，直接修改训练数据的方案不被允许，且严重影响模型训练效果；二是数据加密的防御方法需要修改模型训练代码，数据沙箱场景中难以实现所有 AI 模型训练代码的逐一修改。

另一类是模型修改方法[11-14]，通过修改模型参数来防止数据泄露。该类方法是数据沙箱模式下一种可行的防御方法。然而目前的防御手段难以在数据沙箱模式下取得良好的防御效果，其原因有三点：一是此类方法主要针对模型本身可能夹带的原始数据信息的问题，并且没有考虑攻击者在模型中主动隐藏数据的情况；二是现有的修改模型梯度或参数的方法需要深入解析 AI 模型训练代码，数据沙箱场景中解析所有代码的人力开销巨大，难以实现；三是缺乏一种检测恶意模型的方法，单纯地修改模型可能影响模型的训练效果。

针对上述问题，本文的研究工作如下。

1) 本文分析了数据沙箱模式下的 2 种攻击手段（即数据处理阶段的攻击和模型训练阶段的攻击），构建 2 种攻击手段的威胁模型。在数据沙箱场景下真实实现了深度学习模型数据窃取的攻击过程，量化分析了 2 种攻击的危害程度和鉴别特征。

2) 本文通过分析 2 种攻击手段的特征，提出了针对性的防御方案：针对数据处理阶段的攻击，本文提出基于模型剪枝的数据泄露防御方法，减少数据泄露量；针对模型训练阶段的攻击，本文提出基于模型参数分析的攻击检测方法，从而拦截恶意模型防止数据泄露。这 2 种方法不需要修改或加密数据，也不需要人工分析深度学习模型训练代码，能够更好地应用于数据沙箱模式下数据窃取防御。

3) 本文实现了数据处理阶段的防御方法和模型训练阶段的攻击检测方法。在实验评估阶段，本文分别在图片分类及人脸识别任务中，验证了防御方法和检测方法的有效性。实验结果表明，本文设计的防御方法能够减少 73%的数据泄露，而检测方法能够有效识别 95%以上的攻击行为。

2 相关工作

2.1 深度学习数据窃取攻击

深度学习数据窃取攻击，也称模型反转攻击，是指攻击者利用模型的参数、中间数据结果或模型预测结果来恢复训练数据中的重要信息，进而达到窃取训练数据的目的，文献[3,7,15]等都属于此类攻击。Carlini 等[3-4]证明了参数数量庞大的语言模型会存储和泄露训练数据，并提出了一种简单有效的数据窃取方法，该方法仅通过使用数据测试语言模型就能够提取原始的训练数据序列。Zhang 等[5]和 Hitaj 等[16]提出了一种利用生成对抗网络（GAN, generative adversarial network）的数据窃取攻击，即通过 GAN 来学习先验知识，帮助实现模型反转

并窃取数据，该攻击方式具有较高的普适性，可用于语言模型[17]。文献[6,18-19]提出了一种联邦学习模式下的数据窃取攻击方法，该方法利用联邦学习模型的梯度，首先生成一对随机的“假”输入和标签，然后从模型中获得假数据的梯度，从而执行了模型反转攻击。通过不断对伪输入和伪标签的优化，伪梯度和真梯度之间的距离达到最小，伪数据更接近原始数据。Song 等[7]提出 3 种利用深度学习模型窃取训练数据的方法。第一种是直接在模型参数的最低有效位编码训练数据集的敏感信息；第二种是利用恶意正则化项使模型参数与敏感信息相关联，或用模型参数符号对敏感信息进行编码；第三种是利用数据处理过程中的数据增强技术生成恶意训练数据，并用恶意数据的标签对敏感信息进行编码。这 3 种攻击允许攻击者在模型中主动隐藏隐私信息，造成巨大数据泄露危害。

在数据沙箱模式下，深度学习数据窃取攻击仍然可以实现。例如文献[7]所提出的攻击手段能够有效绕过数据沙箱对训练数据的隐私保护机制，实现数据窃取的目的。目前，国内外研究尚缺乏对数据沙箱模式下深度学习数据窃取攻击的详细威胁分析和实验验证。

2.2　数据窃取攻击防御手段

针对上述数据窃取攻击，国内外学者提出的防御手段主要分为两大类：一类是数据修改方法，通过在原始数据中添加扰动或采用加密技术来保护训练数据；另一类是模型修改方法，通过修改模型参数来防止数据泄露。具体相关工作如下。

1) 数据修改方法

Zhang[8]提出了一种向原始数据中添加随机噪声或用新样本扩展数据集的方法，该方法隐藏单个样本的属性或一组样本的统计特性的敏感信息。文献[9-10, 20]提出了使用同态加密技术实现隐私数据的加密计算，使数据分析人员在不接触原始数据的前提下实现模型训练或数据分析。闫玺玺等[21]则采用区块链实现数据搜索过程中的隐私保护。进一步研究中，Zhang 等[22]提出了一种直接在密文上训练深度学习模型的完全同态加密方案。Rahulamathavan 等[23]提出了一种利用 Paillier 加密系统将 SVM 决策函数转换为密文计算的方案，该方案中测试数据也被加密，所有计算都在密文上进行。文献[24-25]分别在数据发布和数据传输过程中加密数据，设计了能够防御数据窃取攻击的系统架构。

上述数据修改的防御方法虽然取得了一定进展，但难以直接应用于数据沙箱模式中。其原因有两点：一是数据沙箱模式下，原始训练数据属于数据拥有者，直接修改训练数据的方案不被允许，且严重影响模型训练效果；二是数据加密的防御方法需要修改模型训练代码，数据沙箱场景中难以实现所有数据挖掘代码的针对性修改。

2) 模型修改方法

文献[11-14]提出了可以通过修改模型的梯度、参数或输出结果来保护训练数据隐私信息。Abadi 等[11]提出了一种随机梯度下降算法，利用差分隐私对模型参数的梯度进行加噪，保证了模型参数不会暴露太多隐私。Golatkar 等[12]通过添加噪声来修改模型参数，以去除关于特定训练数据集的信息。Jia 等[13]提出在模型的输出中以一定的概率加入噪声，用于预防成员推断攻击，其效果能够使成员推断攻击的成功率降低到 50%。另外，为了防止训练数据的模型记忆过多数据，可以对模型训练算法本身进行修改[18, 26-27]。如 Cao 等[26]提出了一种将模型学习算法转化为求和的形式来遗忘训练数据的方法；Salem 等[18]提出了一种基于模型叠加的防御方法，用于防御对机器学习模型的攻击，以避免单个模型对训练数据的过度记忆。针对边缘计算和联邦学习的新场景，需要在参数聚合的过程中增加噪声实现隐私保护[28]。

通过解析模型参数来预防数据窃取，是数据沙箱模式下一种可行的防御方法。然而目前的防御手段难以在数据沙箱模式下取得良好的防御效果，其原因有三点：一是上述方法主要针对模型本身可能导致的隐私泄露，并且没有考虑攻击者在模型中主动隐藏数据的情况；二是现有的修改模型梯度或参数的方法需要深入解析模型训练代码，数据沙箱场景中，逐一解析代码的人力开销巨大，难以实现；三是缺乏一种检测恶意模型的方法，单纯地修改模型可能影响模型的训练效果。

3　深度学习数据窃取攻击分析

3.1　威胁模型

本文构建了数据沙箱模式下的深度学习数据窃取攻击威胁模型，如图 1 所示。

正常深度学习模型训练过程。正常数据分析人

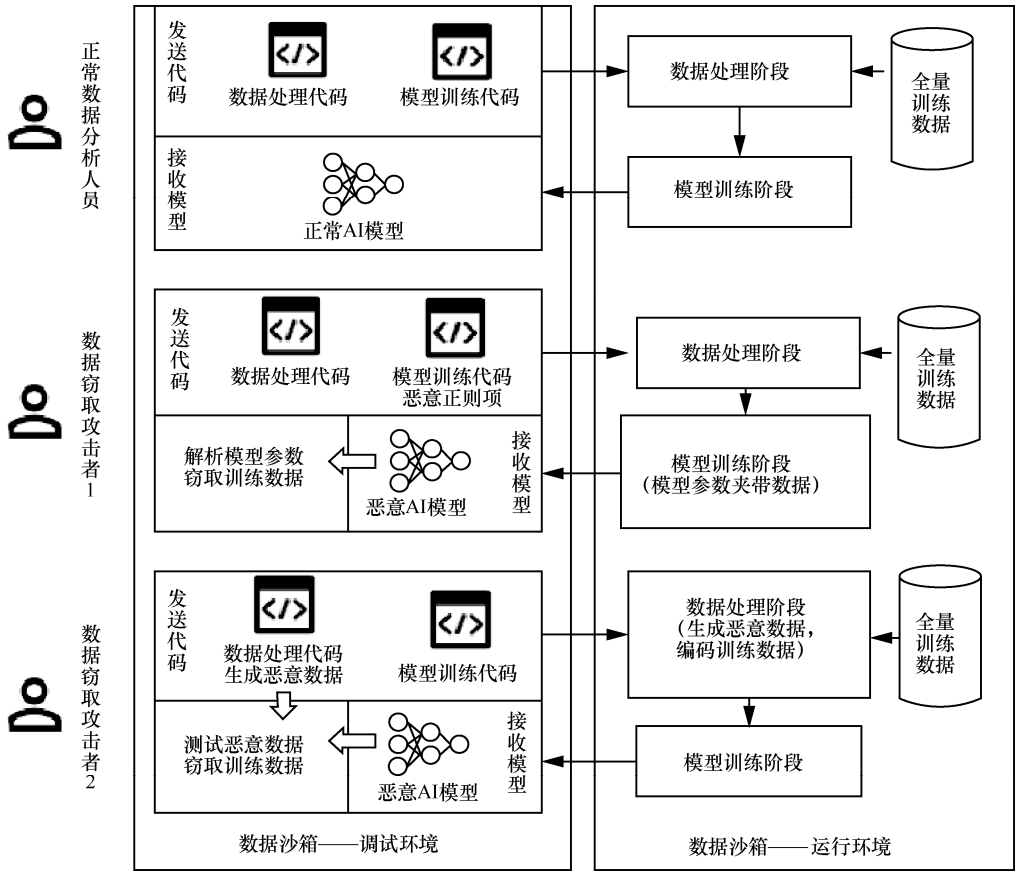

图 1　数据沙箱模式下的深度学习数据窃取攻击威胁模型

员会在数据沙箱的调试环境中根据脱敏的样例数据编写数据处理代码和模型训练代码。代码调试完成后，数据分析人员会将其发送到数据沙箱的运行环境。在运行环境中，数据处理代码将会在全量训练数据 D 中运行，得到处理后的训练数据 D'。该过程可能包含数据增强操作用于扩充数据量、提高训练效果，数据增强操作得到的数据为 D_{A}，而 D'是 D 和 D_{A} 的并集，本文定义 D'如式(1)所示。

$$D'=\{(x_i,y_i)\}_{i=1}^{n},1\leqslant i\leqslant n \tag{1}$$

其中，n 是数据集 D'中的数据条目个数，x_i 是每条数据的特征值（例如图片分类任务中的图片像素），y_i 是每条数据的标签值（例如图片分类任务中的图片种类）。AI 模型可以定义为参数为 $\boldsymbol{\theta}$ 的函数 $f_{\boldsymbol{\theta}}$，参数 $\boldsymbol{\theta}$ 中包含实数的个数为 k，本文定义模型优化过程如式(2)所示。

$$\min_{\boldsymbol{\theta}}\sum_{i=1}^{n}L(y_i,f_{\boldsymbol{\theta}}(x_i))+\Phi \tag{2}$$

其中，函数 L 为深度学习的损失函数，用于评价函数 $f_{\boldsymbol{\theta}}$ 对 x_i 的判断结果与真实标签值 y_i 之间的差距。$\Phi(\theta)$ 是正则项，通常用于防止 AI 模型出现过拟合的情况。AI 模型训练优化的过程可理解为根据训练数据 D'不断优化函数 $f_{\boldsymbol{\theta}}$，进而缩小损失函数的过程。模型训练完成后，数据分析人员可以从数据沙箱中提取训练好的模型 $f_{\boldsymbol{\theta}}$，用于具体 AI 判断任务。

数据沙箱模式下，正常数据分析人员仅能通过样例数据编写代码而无法接触全量数据，因此无法直接拷贝数据或窃取其中的关键信息。然而，在运行环境的数据处理阶段和模型训练阶段，分析人员的代码直接作用于全量数据，因此仍然存在数据泄露的可能。

数据窃取攻击 1。该攻击手段主要作用于模型训练阶段，攻击者恶意修改模型训练过程中的正则项 $\Phi(\boldsymbol{\theta})$，本文定义其为 $\Omega(\boldsymbol{\theta})$。$\Omega(\boldsymbol{\theta})$ 使深度学习模型能够夹带训练数据的原始数据信息。接收到模型 $f_{\boldsymbol{\theta}}$ 后，攻击者能够从参数 $\boldsymbol{\theta}$ 中恢复原始训练数据。其具体效果与攻击过程在 3.2 节中详细阐述。

数据窃取攻击 2。该攻击手段主要作用于数据处理阶段，攻击者恶意修改数据处理过程中的数据增强函数，本文定义恶意数据 D_{M} 如式(3)所示。

$$D_{\mathrm{M}}=\{(x_j^{\mathrm{M}},y_j^{\mathrm{M}})\}_{j=1}^{m},\ 1\leqslant j\leqslant m \tag{3}$$

恶意数据增强函数使 D_{M} 的特征值 x_j^{M} 具有一定的生成规律且不依赖训练数据，而 D_{M} 的标签值 y_j^{M} 中能够夹带原始数据信息。接收到模型 $f_{\boldsymbol{\theta}}$ 后，攻击者重新生成恶意的 x_j^{M}，并用模型 $f_{\boldsymbol{\theta}}$ 测试 x_j^{M}，进而复原标签值 y_j^{M}，从中提取原始数据信息。其具体效果与攻击过程在 3.3 节中详细阐述。

3.2　模型训练阶段的攻击

第一种针对模型训练阶段的攻击，首先要从训练数据的特征值中抽取实数串 $\boldsymbol{s}$，该串中的元素个数与模型参数 $\boldsymbol{\theta}$ 中参数的个数相同，为 k 个。而在训练过程中，攻击者将正则项修改为 $\Omega_1(\boldsymbol{\theta},\boldsymbol{s})$，如式(4)所示。

$$\Omega_1(\boldsymbol{\theta},\boldsymbol{s})=-\alpha\frac{\left|\sum_{i=1}^{k}(\theta_i-\bar{\theta})(s_i-\bar{s})\right|}{\sqrt{\sum_{i=1}^{k}(\theta_i-\bar{\theta})^2}\sqrt{\sum_{i=1}^{k}(s_i-\bar{s})^2}} \tag{4}$$

其中，$\bar{\theta}$ 和 $\bar{s}$ 是指 $\boldsymbol{\theta}$ 和 $\boldsymbol{s}$ 中所有元素的平均值，参数 α 控制了恶意正则项对训练过程的影响。该方法中恶意正则设计采用了皮尔森相关系数衡量模型参数 $\boldsymbol{\theta}$ 与窃取数据 $\boldsymbol{s}$ 之间的相关程度。训练过程中，参数 $\boldsymbol{\theta}$ 与数据 $\boldsymbol{s}$ 相关程度变大时，损失函数的数值将会减小，进而使参数 $\boldsymbol{\theta}$ 向与数据 $\boldsymbol{s}$ 相同的趋势发展。训练结束后，攻击者只要提取 $\boldsymbol{\theta}$ 则可窃取训练数据的特征值。

第二种针对模型训练阶段的攻击，利用符号编码训练数据的特征值。例如训练数据是图片（包含[0, 255]的像素点），该攻击方法则将图片中一个像素点编码为 $\boldsymbol{s}\in\{-1,1\}^k$ 表示的 8 位字符串。能够编码的字符串总长度等于参数 $\boldsymbol{\theta}$ 中元素的个数 k。该编码方式下，攻击者将训练过程的正则项修改为 $\Omega_2(\boldsymbol{\theta},\boldsymbol{s})$，如式(5)所示。

$$\Omega_2(\boldsymbol{\theta},\boldsymbol{s})=-\alpha\frac{\sum_{i=1}^{k}\left|\max(0,-\theta_i s_i)\right|}{k} \tag{5}$$

其中，参数 α 用于控制恶意正则项对训练过程的影响。当 θ_i 与 s_i 符号不相同时，$|\max(0,-\theta_i s_i)|$ 的取值为正值，从而增大损失函数的数值，使 $\boldsymbol{\theta}$ 和 $\boldsymbol{s}$ 中符号相同的方向发展。训练结束后，攻击者提取参数 $\boldsymbol{\theta}$ 的符号，每 8 个元素解码出一个像素值，进而还原图片。

模型训练阶段的攻击强度主要与模型参数个数有关。例如用包含 1 100 万个参数的 PreActResNet18 网络窃取 CIFAR10 数据集中 32×32 的灰度图片数据，第一种攻击方式可以一次性窃取图片 1.07 万张，第二种攻击方式可以一次性窃取图片 1 340 张。

3.3　数据处理阶段的攻击

以窃取 CIFAR10 数据为例，数据处理阶段的攻击过程如图 2 所示。首先，提取图片数据中的像素值并对其进行压缩，例如将[0, 255]的像素值点压缩到[0, 15]，这一步的目的在于用更少的恶意数据编码更多的训练数据。由于 CIFAR10 数据是 10 分类任务，因此生成数据标签值的取值范围为[0, 9]，即编码[0, 15]的像素值数据需要用 2 个标签来编码一个像素点。图 2 中，本文使用标签值 y_1 与 y_2 之和来编码 P_{11} 的一个像素点，使用恶意数据生成算法，分别生成图片 x_1 与 x_2 来匹配标签值。恶意数据生成算法可以采用任意的伪随机数据生成方法，为实现更明显的攻击效果，本文采用了算法 1 所示的恶意数据生成算法。

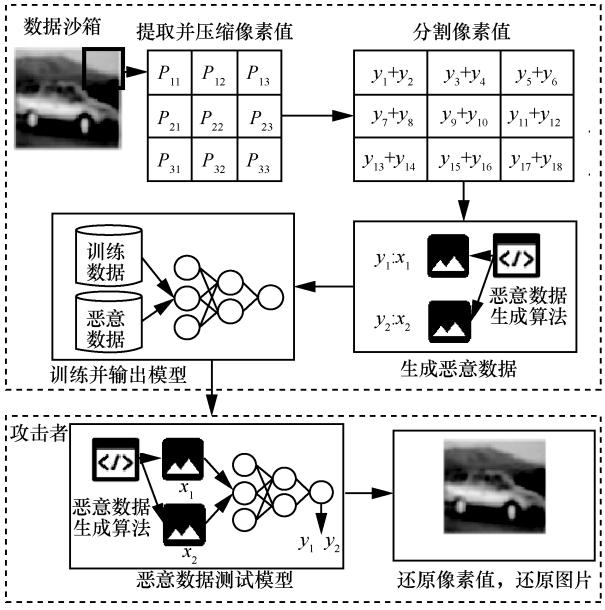

图 2　数据处理阶段的攻击过程

算法 1　恶意数据生成算法

定义编码的图像编号为 u，编码的像素点为 P_{ij}，单张图片的高为 H、宽为 W、通道数为 C、单通道像素数量为 $N=H\times W$，初始化恶意图片 x_1 和 x_2 为

$C\times N$ 的全 0 矩阵；

1) $x_1[u \bmod C][i\times j] = u/3 + 1$;

2) $x_1[u \bmod C][i\times j+1 \bmod N] = 1$;

3) $x_2[u \bmod C][i\times j] = u/3 + 1$;

4) $x_2[u \bmod C][i\times j+1 \bmod N] = 2$;

5) 将 x_1 转化为 $H\times W\times C$ 的矩阵;

6) 将 x_2 转化为 $H\times W\times C$ 的矩阵;

7) 输出恶意图片 x_1 和 x_2;

生成恶意数据 D_{M} 后，将其与原始训练数据融合，训练 AI 模型 $f_{\boldsymbol{\theta}}$。多次反复训练后，模型 $f_{\boldsymbol{\theta}}$ 在恶意数据上的分类准确率达到较高水平（本文实验可达到 100%）。攻击者从数据沙箱中取得 AI 模型 $f_{\boldsymbol{\theta}}$，在本地应用同样的恶意数据生成算法生成 D_{M}，并将其输入 $f_{\boldsymbol{\theta}}$ 中，得到编码了原始训练数据的标签值，进而恢复原始训练数据。

数据处理阶段的攻击强度主要与数据编码方式有关。窃取 CIFAR10 数据集中压缩为 32×32 的灰度图片数据，每编码 2 048 张恶意图片可以窃取一张原始数据。

本节实现的 3 种攻击恢复图片效果如图 3 所示。从图 3 可以看出，3 种攻击均能在数据沙箱场景下有效窃取原始训练数据。

(a) CIFAR10原始数据 (b) 模型训练阶段攻击1恢复数据

(c) 模型训练阶段攻击2恢复数据 (d) 数据准备阶段攻击恢复数据

图 3 深度学习数据窃取攻击实际效果

4 深度学习数据窃取检测与防御方法

数据沙箱模式下，利用深度学习模型窃取数据的攻击具有隐蔽性，而监管者人工审核代码工作量巨大。本文从训练过程和输出模型角度入手进行了详细分析，并设计了面向深度学习模型本身的防御和检测方法。

4.1 深度学习数据窃取攻击特征分析

首先，本文对数据窃取攻击中训练的模型参数进行了详细分析。正常训练过程中，深度学习损失函数的正则项一般选取 L1 范式或 L2 范式，如式(6)和式(7)所示。

$$\Phi(\boldsymbol{\theta}) = \lambda\sum_i |\theta_i| \tag{6}$$

$$\Phi(\boldsymbol{\theta}) = \lambda\sum_i \theta_i^2 \tag{7}$$

L1 范式或 L2 范式的约束下，模型参数的分布通常属于正态分布。恶意模型训练过程引入了与数据或数据编码相关的正则项，因此模型参数的分布可能发生变化。例如第一种针对模型训练阶段的攻击，恶意正则使模型参数分布与图像的像素值的分布近似，而图像的像素值分布具有一定规律，通常不同于正态分布。

本文在 CIFAR10 数据集上，分别训练了多个正常模型和攻击模型，抽取其中一个卷积层的参数进行统计分析，得到了如图 4 所示的结果。在图 4 中，4 幅图的左图分别代表了 4 种情况下多次模型训练的平均参数分布情况，右图分别代表了多次模型训练中各次参数分布情况。通过分析图 4 结果发现模型训练阶段的攻击导致模型参数分布规律异常，而数据处理阶段的攻击没有造成参数的明显分布变化。

数据处理阶段的攻击难以从参数分布的角度进行判断。由于该攻击中恶意数据与真实训练数据的特征不同，本文考虑 2 种数据在训练过程中对隐藏神经元激活分布可能存在差异。针对这一假设，本文进行了模型隐藏层神经元的激活值分析。首先，复现数据处理阶段的攻击过程，并得到其模型参数。然后，将原始训练数据和恶意数据分别输入模型中进行处理，提取隐藏层神经元的激活值。为直观显示恶意数据与真实训练数据对神经元的激活值的影响，本文对提取的神经元激活值进行了降维分析，其过程分别采用了 4 种数据降维算法，包括主成分分析（PCA, principal component analysis）、KernelPCA、t-SNE 和多维标度分析（MDS, multi-dimensional scaling）。神经元激活值的降维分析结果如图 5 所示。从图 5 中可以直观看到，正常数据和恶意数据对隐藏层激活值域的影响存在差异。进一步采用支持向量机（SVM, support vector machine）分析降维的激活值，则 CIFAR10 实验中正确分类正常数据和恶意数据的平均概率为 96%，Olivetti 任务中正确分类平均概率趋近于 100%。

针对模型训练阶段攻击导致参数异常的特点，

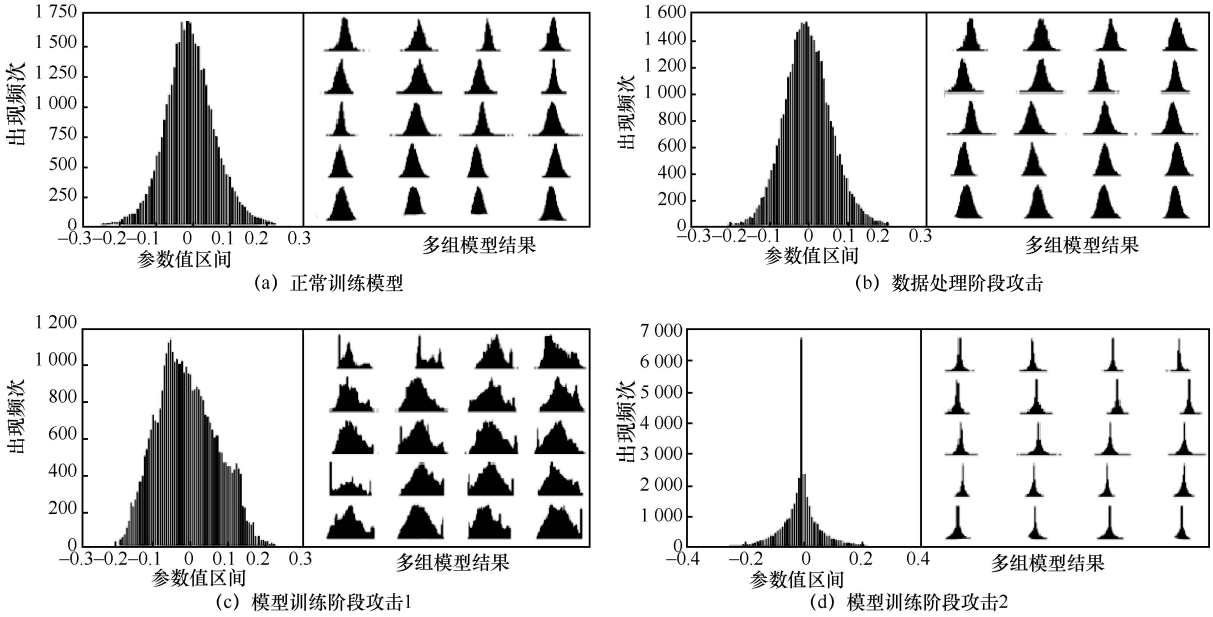

图 4　深度学习数据窃取攻击的模型参数分析结果

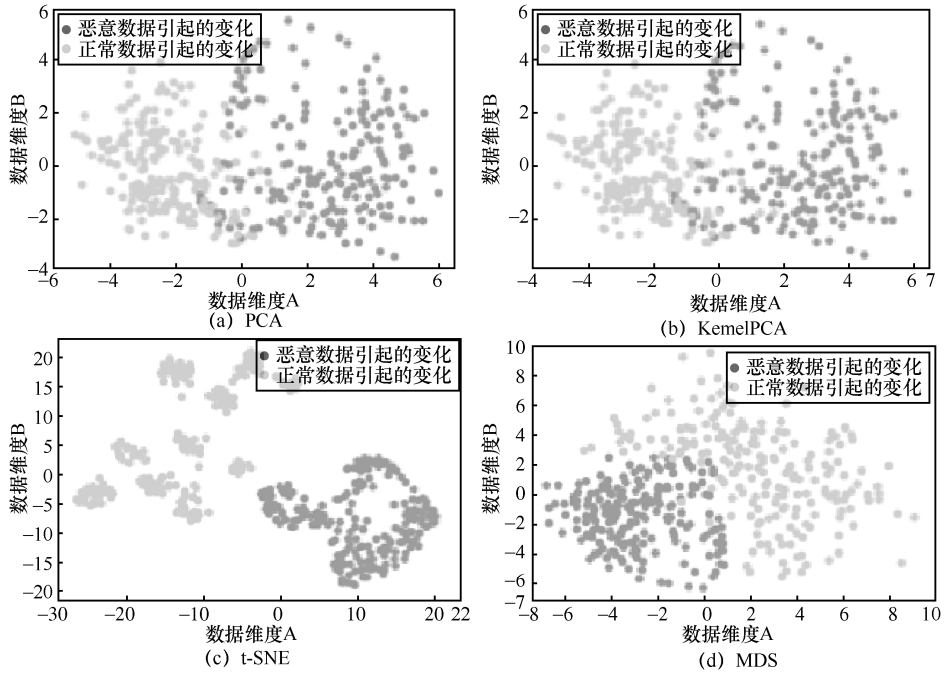

图 5　神经元激活值的降维分析结果

本文设计了 4.2 节所示的攻击检测方法。针对据处理阶段攻击的特殊神经元激活模式，本文设计了 4.3 节所示的模型剪枝防御手段。

4.2　模型训练阶段攻击的参数分析检测

针对恶意训练模型和正常训练模型之间的参数分布差异，本文提出提取参数关键特征值的方法自动化区分恶意模型与正常模型。

对任意输入的模型参数 $\boldsymbol{\theta}$，本文首先统计分析其中某一卷积层参数的分布特征，主要包括参数的极小值、极大值、数量、均值、方差等。为评判其中模型参数 $\boldsymbol{\theta}$ 是否符合正态分布，本文额外引入了偏度（S）和峰度（K）2 个统计概念，两者计算方式如式(8)和式(9)所示。

$$S(\boldsymbol{\theta}) = \mathrm{E}\left[\left(\frac{\boldsymbol{\theta}-\bar{\theta}}{\sigma}\right)^3\right] \tag{8}$$

$$K(\boldsymbol{\theta})=\mathrm{E}\left[\left(\frac{\boldsymbol{\theta}-\bar{\theta}}{\sigma}\right)^4\right] \tag{9}$$

其中，$\bar{\theta}$ 代表参数的均值，σ 代表参数的标准差。从图像上看，偏度表示参数分布的中心位置是否偏离正态分布的中轴线，而峰度表示参数分布中心值的尖端高度。

提取分布特征后，本文利用多组深度学习模型的实际分布特征训练了多个分类器，包括逻辑回归模型、随机森林模型、SVM、Adaboost 和 XGBoost。上述分类器的效果将在第 5 节实验分析部分进行评价和比较。

4.3 数据处理阶段攻击的模型剪枝防御

分析图 5 所示的神经元激活情况可知，恶意窃取数据的模型和常规训练模型，在激活神经元分布上存在差异。理论上，剪除夹带恶意数据的神经元同时保留其他神经元，就能够在防御数据窃取攻击而不损失模型准确率。已有理论证明模型剪枝技术能够有力预防后门数据投毒攻击[29]。该方法的思路是定位到剪除后门数据预测过程中激活值最大的神经元，进而降低后门数据预测的准确率。

本文参考这种方法，在数据沙箱场景下进行了改进。该防御方法主要思想是：用正常数据测试模型隐藏层的神经元激活值，对于那些正常测试数据预测过程中激活值较小神经元进行剪除，这种剪除对深度学习模型原本任务影响较小，而更有可能夹带训练数据。筛选剪除神经元的具体步骤如下。

1) 对于模型 $\boldsymbol{\theta}$，输入该任务相关的测试数据 D_{test}，在模型处理数据的过程中，提取隐藏层每一个神经元 θ_i 的激活值序列 $\boldsymbol{w}_i$。

2) 对所有 $\boldsymbol{w}_i$ 的序列求平均值 $\bar{w}_t$，并将隐藏层神经元按照 $\bar{w}_t$ 的绝对值 $|\bar{w}_t|$ 由小到大排序。

3) 将 $|\bar{w}_t|$ 值排在最小的 n 个神经元设置为 0 值，即实施剪除操作。

4) 重复上述操作步骤，直到剪除神经元个数达到设置值 N。

在 5.3 节的实验中，本文对该模型剪枝防御手段进行了实验评价。

5 实验分析

5.1 实验设置

实验环境。本文实验是在 12 核 2.10 GHz Intel（R）Xeon（R）Silver 4116 CPU、128 GB RAM 和 8 GB NVIDIA Quadro P4000 GPU 的计算机上进行的，并使用 Pytorch 框架实现攻击和防御方法。

实验数据。本文采用图片分类任务数据集 CIFAR10 数据集和人脸识别任务 Olivetti 数据集进行实验。其中 CIFAR10 数据集包含 60 000 张彩色图像，这些图片共分为 10 类，像素值为 32×32×3。Olivetti 人脸数据集包含 400 张人脸灰度图，分别来自具有不同特征的 40 人，每张图的分辨率为 64 dpi×64 dpi。实验数据的测试集和训练集划分为 75%和 25%。

AI 模型。为完成图片分类和人脸识别任务，本文使用了 ResNet34 和 PreActResNet-18 模拟攻击过程，其中 ResNet34 网络包含参数约 46 万，而 PreActResNet18 网络包含参数 1 100 万。

5.2 攻击危害程度评估实验

本文采用准确率和宏平均 F1 值来评价攻击的危害程度以及原始模型的效果。准确率指模型预测正确的数据占全量数据的比例；宏平均 F1 值，又称 macro-F1 值，该值综合考虑多分类任务中每一类分类数据的 F1 值（F1 值是精确率和召回率的调和平均数），用于评价多分类任务的整体效果。

针对第一种模型训练攻击，本文额外采用相关系数 P 来评价窃取数据和原数据的相似程度，该值由式(4)所示的皮尔森相关系数求出。针对第二种模型训练攻击，本文额外采用符号相关度 Q 来评价窃取数据和原数据的相似程度，该值统计了模型参数 $\boldsymbol{\theta}$ 与窃取数据 $\boldsymbol{s}$ 中对应为符号相同的元素出现的概率。这 2 种模型训练攻击均用参数 α 控制了恶意正则项对训练过程的影响。针对数据处理过程的攻击，本文分别合成了不同数据量级的恶意数据，实验分析其对原始模型的影响程度以及攻击效果。

模型训练阶段攻击危害程度评估实验数据如表 1 所示。对于模型训练阶段的第一种攻击而言，其窃取图片与原图的相似度在 2 个数据集、2 种模型上均能达到 99%。同时，窃取数据对原本模型训练任务的影响不大。通过调节参数 α，图片分类任务和人脸识别任务的准确率和宏平均 F1 值均能达到 85%以上，因此难以从原始任务效果好坏角度鉴别恶意攻击。对于模型训练阶段的第二种攻击而言，其攻击效果在 CIFAR10 数据集上表现较好，能够达到 90%，而在 Olivetti 数据集上相似度在 70%左右。2 种模型、2 种数据集上的攻击对原始任务效果的影响不大，平均准确率和和宏平均 F1 值在 90%左右。

对于数据处理阶段的攻击如表 2 所示，其窃取

表 1　　**模型训练阶段攻击危害程度评估实验数据**

攻击方式	数据集	AI 模型	原始模型效果		攻击危害程度		
			准确率	宏平均 F1 值	参数 α	相关系数 P	符号相关度 Q
模型训练阶段攻击 1	CIFAR10	ResNet34	0.881	0.881	4.0	0.960	—
			0.861	0.861	8.0	0.980	—
			0.839	0.839	16.0	0.990	—
		PreActResNet18	0.932	0.932	4.0	0.969	—
			0.924	0.924	8.0	0.984	—
			0.901	0.901	16.0	0.991	—
	Olivetti	ResNet34	0.908	0.908	4.0	0.831	—
			0.917	0.917	8.0	0.961	—
			0.883	0.883	16.0	0.992	—
		PreActResNet18	0.900	0.900	4.0	0.993	—
			0.867	0.867	8.0	0.999	—
			0.881	0.881	16.0	0.960	—
模型训练阶段攻击 2	CIFAR10	ResNet34	0.926	0.926	8.0	—	0.845
			0.922	0.922	16.0	—	0.949
			0.917	0.917	32.0	—	0.984
		PreActResNet18	0.941	0.941	8.0	—	0.764
			0.946	0.946	16.0	—	0.853
			0.946	0.946	32.0	—	0.910
	Olivetti	ResNet34	0.900	0.900	8.0	—	0.536
			0.917	0.917	16.0	—	0.579
			0.917	0.917	32.0	—	0.696
		PreActResNet18	0.527	0.891	8.0	—	0.764
			0.908	0.908	16.0	—	0.561
			0.892	0.892	32.0	—	0.632

表 2　　**数据处理阶段攻击危害程度评估实验数据**

攻击方式	数据集	AI 模型	原始模型效果		攻击危害程度		
			准确率	宏平均 F1 值	合成数据量	准确率	宏平均 F1 值
数据准备阶段攻击	CIFAR10	ResNet34	0.924	0.924	8 192	0.991	0.944
			0.920	0.919	18 432	0.999	0.945
			0.913	0.913	38 912	0.999	0.938
		PreActResNet18	0.945	0.983	8 192	0.999	0.999 4
			0.945	0.998	18 432	1.0	1.0
			0.938	0.999	38 912	1.0	1.0
	Olivetti	ResNet34	0.892	0.892	2 048	1.0	1.0
			0.916	0.916	4 096	0.999	0.999
			0.933	0.933	8 192	0.999	0.999
		PreActResNet18	0.966	0.966	2 048	0.989	0.989
			0.900	0.900	4 096	0.998	0.998
			0.891	0.891	8 192	0.999	0.999

图片的效果最佳，在 2 个数据集、2 种模型上均能达到 99%，甚至在 CIFAR10 数据集的 PreActResNet18 模型和 Olivetti 数据集的 ResNet34 模型上能够达到 100%。由数据可见，数据处理阶段的攻击对原本模型训练任务的影响也不明显，其任务平均的准确率在 90%左右。

同样针对 CIFAR10 数据集，一般场景下的数据窃取攻击数据恢复的准确率在 94%～99%（准确率根据文章中的平均绝对像素误差折算），而对原始模型的准确率影响约为 2%[7]。本文将此类攻击有效实现于数据沙箱场景，得到了与之类似的结果。

通过实验分析得出结论，深度学习数据窃取攻击在数据沙箱场景下的危害极大，其特征在于数据窃取的准确率高，且不影响原有模型的训练，难以察觉。

5.3 参数分析检测实验

根据 4.2 节所提方法，本文首先提取了已完成训练模型的 360 组卷积层模型参数，其中包含正常模型 17%、模型训练阶段攻击 1 产生模型 25%和模型训练阶段攻击 2 产生模型 58%。然后，将这些模型参数的极小值、极大值、数量、均值、方差、峰度、偏度等统计特征作为特征值训练机器学习分类器。

逻辑回归模型、随机森林模型、SVM、Adaboost 和 XGBoost 对恶意神经网络的识别准确率和宏平均 F1 值如表 3 所示。实验显示，5 种机器学习分类器对恶意神经网络的识别准确率和宏平均 F1 值均达到 91%以上，且 SVM 的识别准确率最高，能够达到 95%以上。

表 3 不同模型对恶意神经网络的识别准确率和宏平均 F1 值

使用模型	识别准确率	宏平均 F1 值
随机森林模型	0.934	0.934
逻辑回归模型	0.934	0.934
SVM	0.953	0.953
Adaboost	0.934	0.934
XGBoost	0.915	0.915

5.4 模型剪枝防御实验

本文实现的模型剪枝防御实验效果如图 6、图 7 所示。站在防御者的角度，希望在训练数据、测试数据准确率尽可能不受到影响的前提下，降低恶意数据上的识别准确率，进而防止数据处理阶段的攻击。

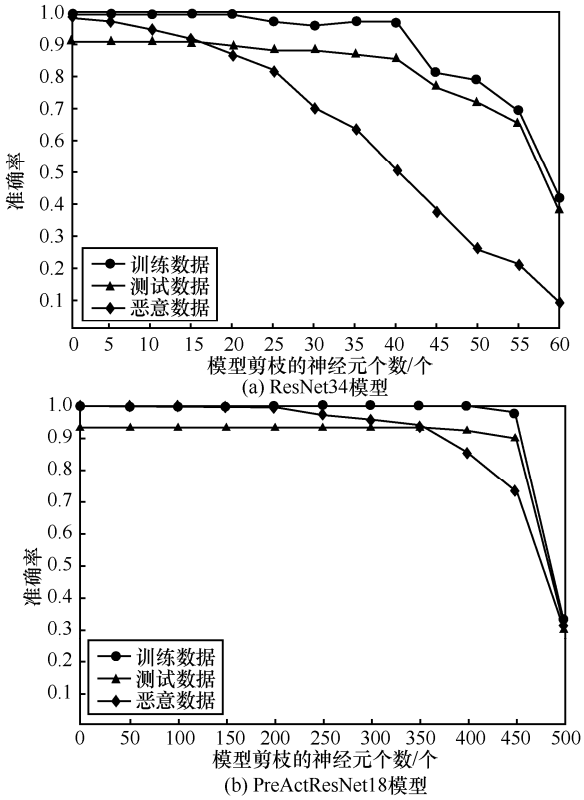

图 6 CIFAR10 数据集模型剪枝防御实验效果

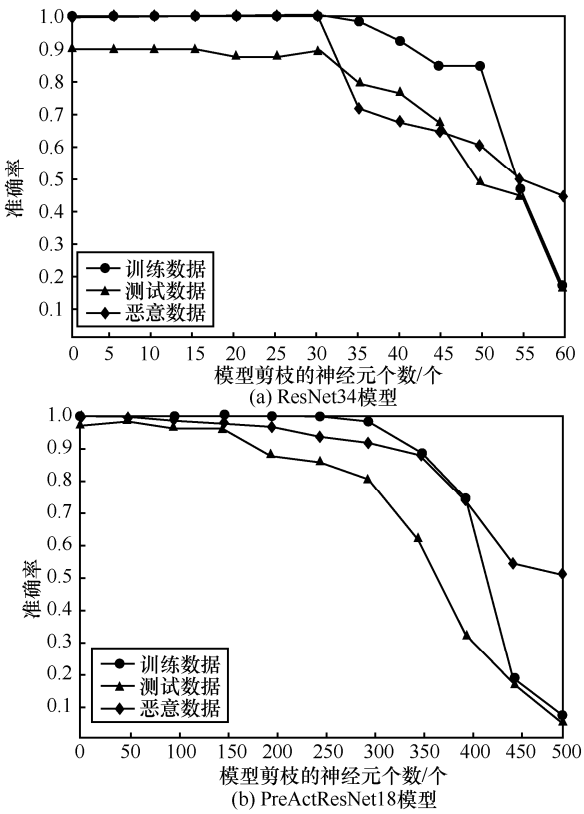

图 7 Olivetti 数据集模型剪枝防御实验效果

如图 6 所示，模型剪枝防御在 CIFAR10 数据集上的 ResNet34 模型中表现最好，当剪除 40 个神经元时，训练数据上的准确率降低了 3%，测试数据上的准确率降低了 7%，而数据防泄露的效果为 73%，而在更加复杂的 PreActResNet18 模型上，选择剪除 450 个神经元能够防止 29%的数据泄露问题。相比于深入分析模型，并在神经元上加入扰动的防御方法（准确率影响为 5%～9%，防御效果为 23%～100%），有类似的防御效果[12]。如图 7 所示，模型剪枝防御在 Olivetti 数据集上的 ResNet34 模型也有良好效果，正确选取模型剪枝神经元个数，能够在模型剪枝防御方法可以保证训练数据、测试数据准确率降低不到 5%的前提下，防止 26%以上数据泄露。

表 4 将本文实现的 2 种防御手段与相关工作中的防御手段进行了对比，可以看出具有恶意模型检测能力和不需要修改训练代码是本文工作的主要贡献点。数据沙箱场景中，原始训练数据属于数据拥有者，直接修改训练数据的方案不被允许，而解析所有 AI 训练代码的人力开销巨大，因此本文工作更加适用于该场景。

表 4　　相关工作对比

相关工作	能否检测恶意模型	是否修改原始数据	是否修改训练代码	直接修改模型参数
文献[18]	×	×	√	×
文献[8-10,22-23]	×	√	×	×
文献[11-14]	×	×	×	√
文献[26-27]	×	×	√	×
本文工作	√	×	×	√

6　结束语

本文分析了数据沙箱模式下数据处理阶段的攻击和模型训练阶段的攻击，构建了 2 种攻击手段的威胁模型。通过在图片分类任务数据集 CIFAR10 数据集和人脸识别任务 Olivetti 数据集上的实验，量化分析了 2 种攻击的危害程度和鉴别特征。

本文通过分析 2 种攻击手段的特征，分别提出了针对性的防御方案：针对数据处理阶段的攻击，本文提出基于模型剪枝的数据泄露防御方法，减少数据泄露量；针对模型训练阶段的攻击，本文提出基于模型参数分析的攻击检测方法，从而拦截恶意模型防止数据泄露。这 2 种方法不需要修改或加密数据，也不需要人工分析深度学习模型训练代码，能够更好地应用于数据沙箱模式下数据窃取防御。

实验评估阶段，本文验证了防御方法和检测方法的有效性。实验结果表明，本文设计的防御方法能够减少 26%～73%的数据泄露，而检测方法能够有效识别 95%以上的攻击行为。

在未来工作中，本文所实现的防御手段尚有 2 项内容需要进一步研究与实现：基于模型参数分析的攻击检测方面，由于应用了机器学习技术检测模型参数，造成检测速度较慢、资源开销较多，需要进一步提升性能；基于模型剪枝的数据泄露防御方面，对于复杂任务复杂模型的防御效果不足，需要设计新算法或新机制提升防御效果。

参考文献：

[1] DELACROIX S, MONTGOMERY J. From research data ethics principles to practice: data trusts as a governance tool[J]. SSRN Electronic Journal, 2020: doi.org/10.2139/ssrn.3736090.

[2] O'HARA K. Data trusts: ethics, architecture and governance for trustworthy data stewardship[R]. 2019.

[3] CARLINI N, LIU C, ERLINGSSON Ú, et al. The secret sharer: evaluating and testing unintended memorization in neural networks[C]// Proceedings of the 28th USENIX Security Symposium. Berkeley: USENIX Association, 2019: 267-284.

[4] CARLINI N, TRAMER F, WALLACE E, et al. Extracting training data from large language models[C]//Proceedings of the 30th USENIX Security Symposium. Berkeley: USENIX Association, 2021: 2633-2650.

[5] ZHANG Y H, JIA R X, PEI H Z, et al. The secret revealer: generative model-inversion attacks against deep neural networks[C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 250-258.

[6] ZHU L G, LIU Z J, HAN S. Deep leakage from gradients[J]. arXiv Preprint, arXiv: 1906.0835, 2019.

[7] SONG C Z, RISTENPART T, SHMATIKOV V. Machine learning models that remember too much[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 587-601.

[8] ZHANG T W. Privacy-preserving machine learning through data obfuscation[J]. arXiv Preprint, arXiv: 1807.01860, 2018.

[9] BRAKERSKI Z, GENTRY C, VAIKUNTANATHAN V. (Leveled) fully homomorphic encryption without bootstrapping[J]. ACM Transactions on Computation Theory, 2014, 6(3): 1-36.

[10] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[C]//Advances in Cryptology — EUROCRYPT'99. Berlin: Springer, 1999: 223-238.

[11] ABADI M, CHU A, GOODFELLOW I, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2016: 308-318.

[12] GOLATKAR A, ACHILLE A, SOATTO S. Eternal sunshine of the spotless net: selective forgetting in deep networks[C]//Proceedings of

2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2020: 9304-9312.

[13] JIA J Y, SALEM A, BACKES M, et al. MemGuard: defending against black-box membership inference attacks via adversarial examples[C]//Proceedings of 2019 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2019: 259-274.

[14] PAPERNOT N, ABADI M, ERLINGSSON U, et al. Semi-supervised knowledge transfer for deep learning from private training data[J]. arXiv Preprint, arXiv: 1610.05755, 2016.

[15] 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: ACM Press, 2015: 1322-1333.

[16] HITAJ B, ATENIESE G, PEREZ-CRUZ F. Deep models under the GAN: information leakage from collaborative deep learning[C]// Proceedings of 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2017: 603-618.

[17] PAN X D, ZHANG M, JI S L, et al. Privacy risks of general-purpose language models[C]//Proceedings of 2020 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2020: 1314-1331.

[18] SALEM A, BHATTACHARYA A, BACKES M, et al. Updates-leak: data set inference and reconstruction attacks in online learning[C]//Proceedings of the 29th USENIX Security Symposium. Berkeley: USENIX Association, 2020: 1291-1308.

[19] WANG Z B, SONG M K, ZHANG Z F, et al. Beyond inferring class representatives: user-level privacy leakage from federated learning[C]//Proceedings of IEEE INFOCOM 2019 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2019: 2512-2520.

[20] 杨攀, 桂小林, 姚婧, 等. 支持同态算术运算的数据加密方案算法研究[J]. 通信学报, 2015, 36(1): 171-182.
YANG P, GUI X L, YAO J, et al. Research on algorithms of data encryption scheme that supports homomorphic arithmetical operations[J]. Journal on Communications, 2015, 36(1): 171-182.

[21] 闫玺玺, 原笑含, 汤永利, 等. 基于区块链且支持验证的属性基搜索加密方案[J]. 通信学报, 2020, 41(2): 187-198.
YAN X X, YUAN X H, TANG Y L, et al. Verifiable attribute-based searchable encryption scheme based on blockchain[J]. Journal on Communications, 2020, 41(2): 187-198.

[22] ZHANG Q C, YANG L T, CHEN Z K. Privacy preserving deep computation model on cloud for big data feature learning[J]. IEEE Transactions on Computers, 2016, 65(5): 1351-1362.

[23] RAHULAMATHAVAN Y, PHAN R C W, VELURU S, et al. Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud[J]. IEEE Transactions on Dependable and Secure Computing, 2014, 11(5): 467-479.

[24] 于东, 康海燕. 面向时序数据发布的隐私保护方法研究[J]. 通信学报, 2015, 36(S1): 243-249.
YU D, KANG H Y. Privacy protection method on time-series data publication[J]. Journal on Communications, 2015, 36(S1): 243-249.

[25] 韩培义, 刘川意, 王佳慧, 等. 面向云存储的数据加密系统与技术研究[J]. 通信学报, 2020, 41(8): 55-65.
HAN P Y, LIU C Y, WANG J H, et al. Research on data encryption system and technology for cloud storage[J]. Journal on Communications, 2020, 41(8): 55-65.

[26] CAO Y Z, YANG J F. Towards making systems forget with machine unlearning[C]//Proceedings of 2015 IEEE Symposium on Security and Privacy. Piscataway: IEEE Press, 2015: 463-480.

[27] NASR M, SHOKRI R, HOUMANSADR A. Machine learning with membership privacy using adversarial regularization[C]//Proceedings of 2018 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2018: 634-646.

[28] 张佳乐, 赵彦超, 陈兵, 等. 边缘计算数据安全与隐私保护研究综述[J]. 通信学报, 2018, 39(3): 1-21.
ZHANG J L, ZHAO Y C, CHEN B, et al. Survey on data security and privacy-preserving for the research of edge computing[J]. Journal on Communications, 2018, 39(3): 1-21.

[29] LIU K, DOLAN-GAVITT B, GARG S. Fine-pruning: defending against backdooring attacks on deep neural networks[C]//Research in Attacks, Intrusions, and Defenses. Cham: Springer International Publishing, 2018: 273-294.

[作者简介]

潘鹤中（1991− ），男，辽宁本溪人，北京邮电大学博士生，主要研究方向为云安全、数据安全、密码学。

韩培义（1992− ），男，山西吕梁人，博士，哈尔滨工业大学（深圳）助理研究员，主要研究方向为数据安全和隐私保护。

向夏雨（1991− ），男，湖南花垣人，北京邮电大学博士生，主要研究方向为隐私保护、医疗大数据分析。

段少明（1994− ），男，湖南邵阳人，哈尔滨工业大学（深圳）博士生，主要研究方向数据安全和机器学习。

庄荣飞（1992− ），男，福建泉州人，哈尔滨工业大学（深圳）博士生，主要研究方向为数据安全、机器学习安全、隐私保护。

刘川意（1982− ），男，四川乐山人，博士，哈尔滨工业大学（深圳）教授，主要研究方向为云计算与云安全、大规模存储系统、数据保护与数据安全。