Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

DNN模型水印及其鲁棒性评估

硕士研究生 邢凤桐

2023年11月12日

问题回溯

- 总结反思
  - 讲解语速较快，讲述内容概念和内涵不明确
  - 未考虑听众的观感和接受程度，算法细节和原理需要讲述清晰

- 相关内容
  - 2023.04.02 夏志豪《深度神经网络鲁棒性评估方法》
  - 2023.03.12 邢凤桐《深度神经网络模型水印保护方法》
  - 2022.07.24 侯钰斌《神经网络模型测试方法与模型健壮性》

内容提要

- 预期收获
- 题目内涵解析
- 研究背景与意义
- 研究历史与现状
- 知识基础
- 算法原理
  - PLKmark
- 特点总结与工作展望
- 参考文献

背景简介

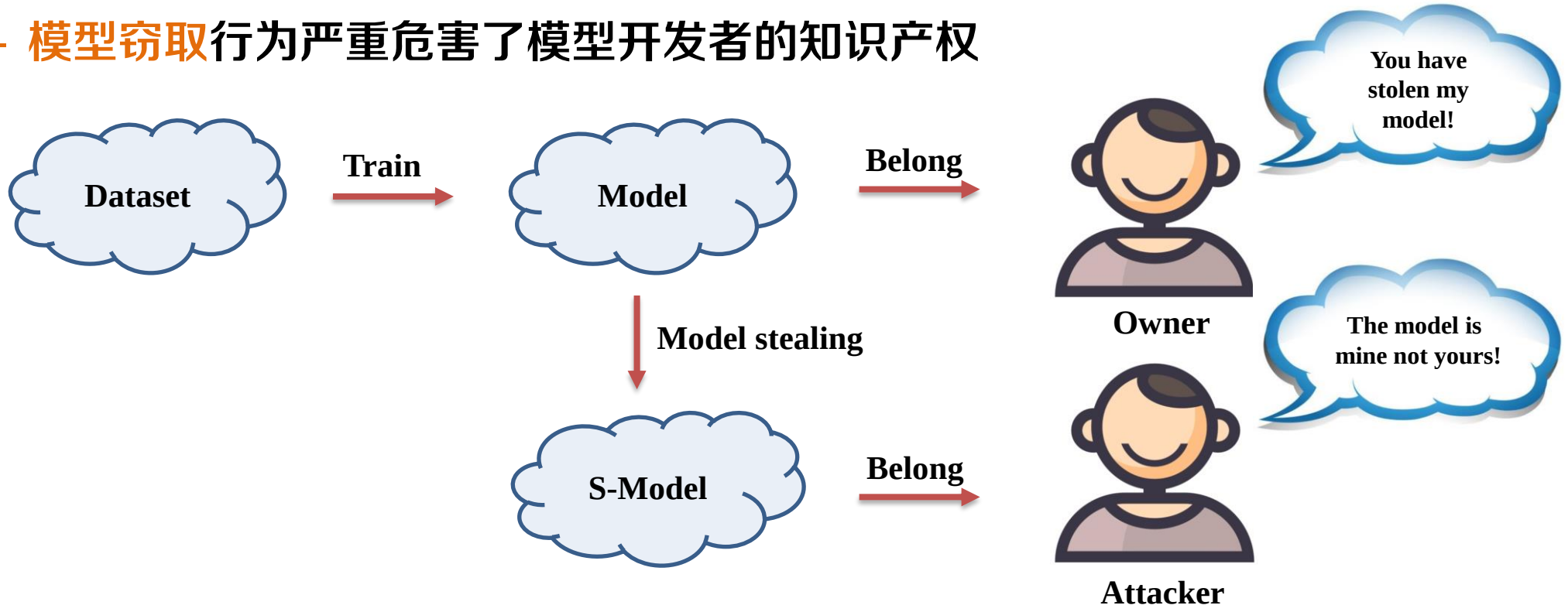
- 预期收获
  - 掌握模型水印基本概念及分类
  - 了解DNN模型水印嵌入及验证方法
  - 了解DNN模型水印鲁棒性评估方法

内涵解析与研究目标

- 题目内涵解析（DNN模型水印及其鲁棒性评估）
  - 水印：一种在数字媒体中嵌入可见或不可见标记的技术
  - 模型水印：一种隐藏在模型中且不影响模型本身功能的特定信息
  - 鲁棒性：系统或模型在面对各种不确定性和干扰时的稳定性和可靠性
  - 鲁棒性评估：对系统或模型的鲁棒性进行量化和测试的过程
  - 模型水印鲁棒性评估：对嵌入在机器学习模型中的水印进行鲁棒性评估的过程
- 研究目标
  - 面向人工智能领域模型知识产权保护
  - 研究模型版权验证、模型水印嵌入、模型水印鲁棒性评估等关键问题
  - 结合深度学习、模型窃取攻击与防御等理论
  - 实现模型水印准确性和鲁棒性的显著提升

研究背景

- 研究背景
  - 深度神经网络技术发展迅速，在图像识别、自然语言处理等领域发挥重要作用
  - 深度神经网络模型的生产成本通常很高，其训练过程繁琐、花销昂贵
  - 商业竞争、非法售卖、模型泄露、数据滥用
  - 模型窃取行为严重危害了模型开发者的知识产权



研究意义

- 研究意义
  - 知识产权保护：防止模型未经授权的复制和使用，确保模型的知识产权不受侵害
  - 模型溯源和验证：提供一种机制来确认模型的真实拥有者和可信性
  - 安全性增强：提高模型对抗攻击的鲁棒性，降低模型遭受恶意攻击的风险

  - 抗数据泄露：对模型的输出结果进行标记，控制数据的滥用和追责
  - 隐私保护：模型水印研究可以探索如何在保护模型知识产权的同时保护用户隐私，力求在保护模型的同时最小化对用户隐私的影响
- 模型水印技术在鲁棒性改善和提升仍存在巨大挑战！

模型水印研究刻不容缓！

研究历史

1997 Cox等人首次提出数字水印的概念，并强调了水印应具备一定的鲁棒性，即使经过压缩、滤波、裁剪等操作，仍能有效地检测和提取出水印信息

2017 Adi等人正式提出模型水印的概念和方法，通过向模型中添加特定的标记信息实现对模型的水印保护，对深度学习模型的安全性提出了一定的挑战，促进了对抗性机器学习领域的研究和发展

2018 Gu等人提出一种通过后门方式为深度神经网络添加水印的方法，通过在模型中插入后门，在特定输入条件下触发水印，从而保护模型的知识产权

2019 Fan等人提出了一种新的基于护照的DNN所有权验证方案，使原始任务的DNN推理性能因伪造护照而显著恶化，对网络修改具有鲁棒性，对歧义攻击具有弹性

2020 Namba等人提出一种查询修改的水印攻击方法，并提出了一种指数加权的水印算法，使得嵌入模型的水印在受到移除攻击时弹性更强

2021 Szyller等人提出了一种DAWN的水印方案，不改变原训练过程，利用API客户端对水印生成对应的唯一标识符，通过模型的输入输出变化分析水印的鲁棒性

2021 Jia等人提出了一种纠缠水印嵌入（EWE）的方法，使用SNNL嵌入纠缠水印，将任务数据和水印数据紧密结合，有效提升了模型水印的鲁棒性，但由于模型不分层不需要纠缠，算法的效率实际上受到影响

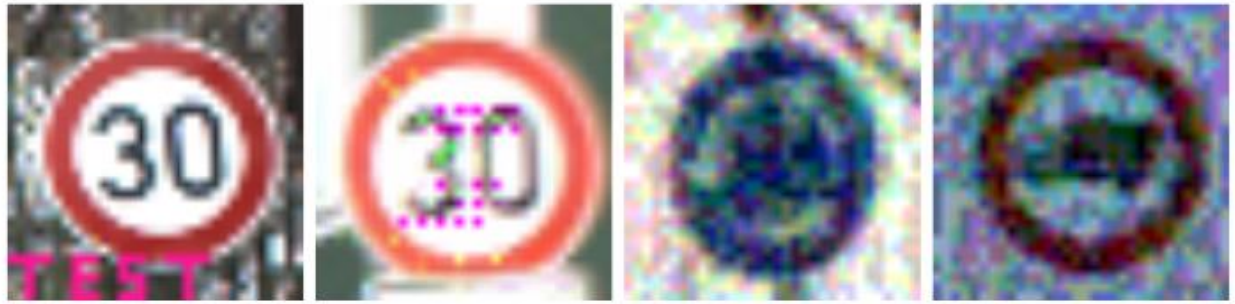
2022 LEE等人通过比较现有的11种模型水印算法在不同攻击下的效果，从性能角度分析和评估了各个水印算法的抗攻击能力，并提出了一种新的自适应攻击方法

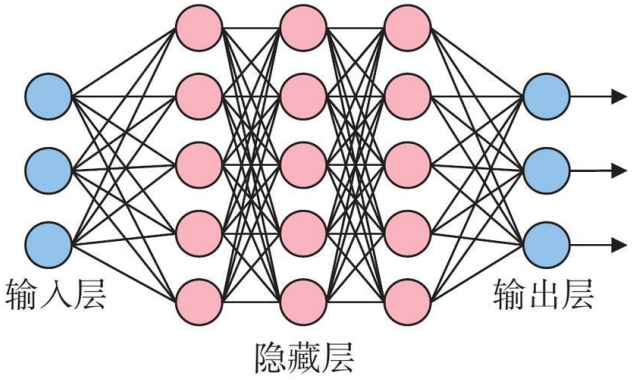
2023 Li等人提出一种强大的PLM黑盒水印框架PLMmark，通过引入了监督对比损失和双重验证，评估并提升了模型水印的鲁棒性

从浅入深，从易入难，模型水印不断从嵌入算法研究转向算法评估与性能提升

研究现状

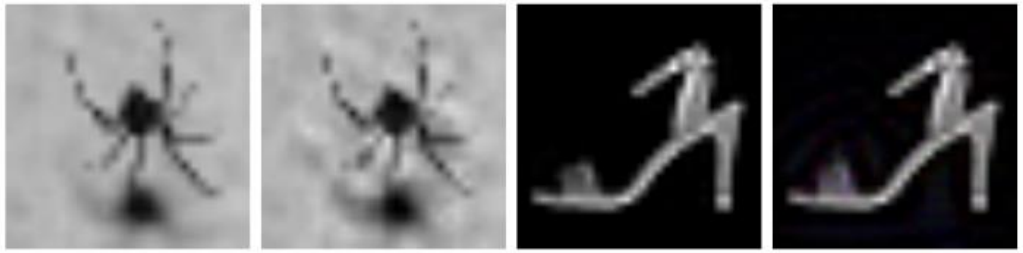
- 模型水印嵌入方法
  - 难以在保持任务数据与水印数据间关联性的同时不影响原模型任务
  - 模型水印嵌入后难以在有效应对水印移除攻击
- 模型水印鲁棒性评估方法
  - 现有的模型水印鲁棒性评估方法仅能评估模型水印的性能鲁棒性，而缺乏对稳定鲁棒性角度的量化评估方法
  - 现有评估方法在不同水印移除攻击情境下的评估结果差异较小





知识基础 模型水印

- 模型水印
  - 基本概念：一种隐藏在模型中且不影响模型本身功能的特定信息
  - 原理：通过修改模型参数（内部结构、输入输出等）让模型过拟合到只有模型所有者知道的异常输入输出关系，用来宣称模型的所有权
  - 目的：防止模型被窃取，保护模型的知识产权
  - 分类
    - 可见水印、不可见水印
    - 白盒水印、黑盒水印、灰盒水印、无盒水印
  - 相关流程
    - 水印生成、水印嵌入、水印提取、水印验证

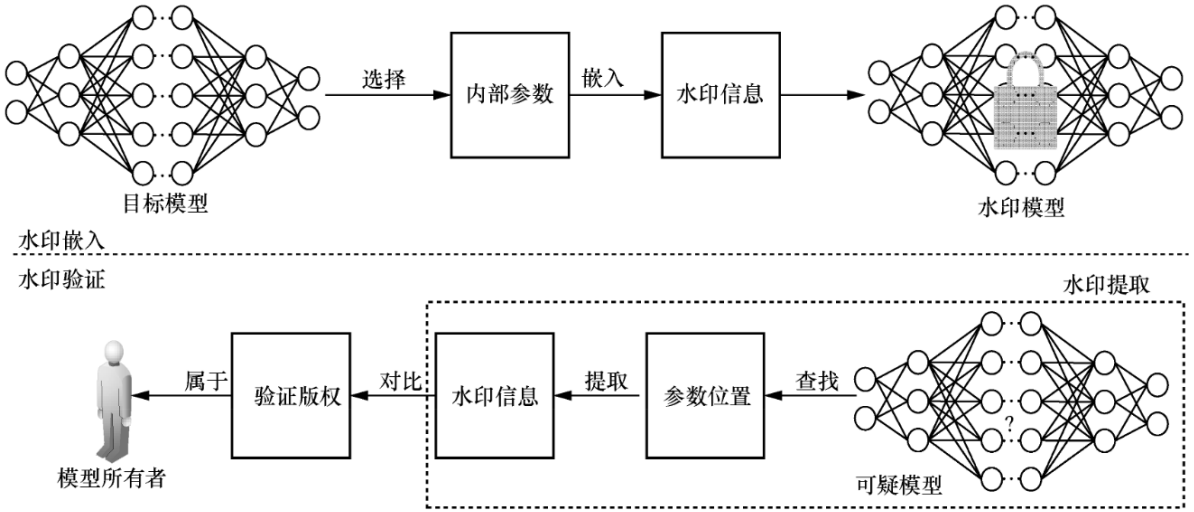


↓

验证所有者对模型的所有权，保护知识产权

知识基础 模型水印分类

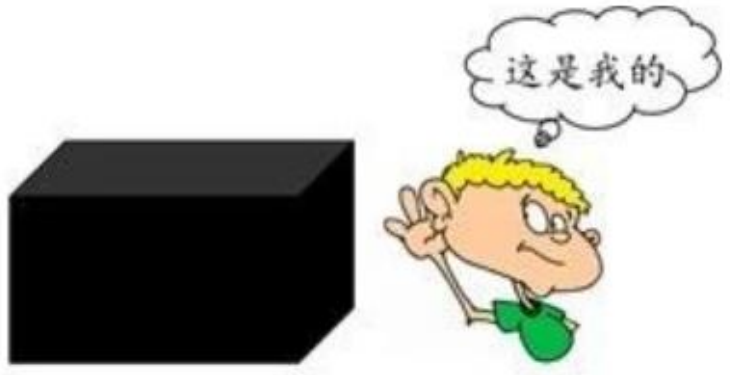
- 白盒水印（White-box Watermarking）
  - 一种注重安全性和隐蔽性的数字水印技术
    - 以加密和密钥管理为基础
  - 特点：安全性和隐蔽性强
    - 通常将水印信息分散嵌入到神经网络模型的不同位置
      - 通过修改已训练好网络的内部信息实现水印的嵌入
    - 通常是不可见或几乎不可见的
    - 提取过程需要特定的解密算法和密钥
  - 局限性
    - 容量限制
    - 鲁棒性与感知质量的平衡
    - 需要针对具体情景的抗攻击能力

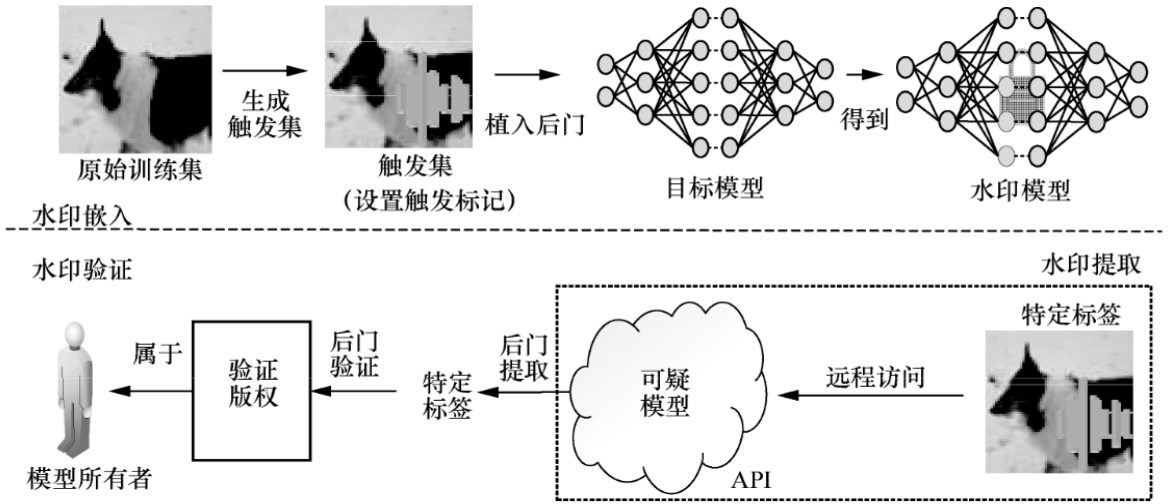


白盒水印注重保护水印信息的安全性和完整性

知识基础 模型水印分类

- 黑盒水印（Black-box Watermarking）
  - 一种注重嵌入和提取效率的数字水印技术
    - 主要用于版权保护和内容追踪
  - 特点：嵌入和提取效率高
    - 不需要大量的计算资源或存储空间
    - 在嵌入和提取过程中算法简单，无需直接访问原始数据的内部结构或特征
      - 特征空间嵌入、参数空间嵌入、算法嵌入
    - 无法获悉神经网络模型的结构和参数
  - 盲提取、扩频水印、选择性水印
  - 局限性
    - 隐蔽性和鲁棒性较弱
    - 依赖于特定环境或系统





黑盒水印主要关注嵌入和提取效率

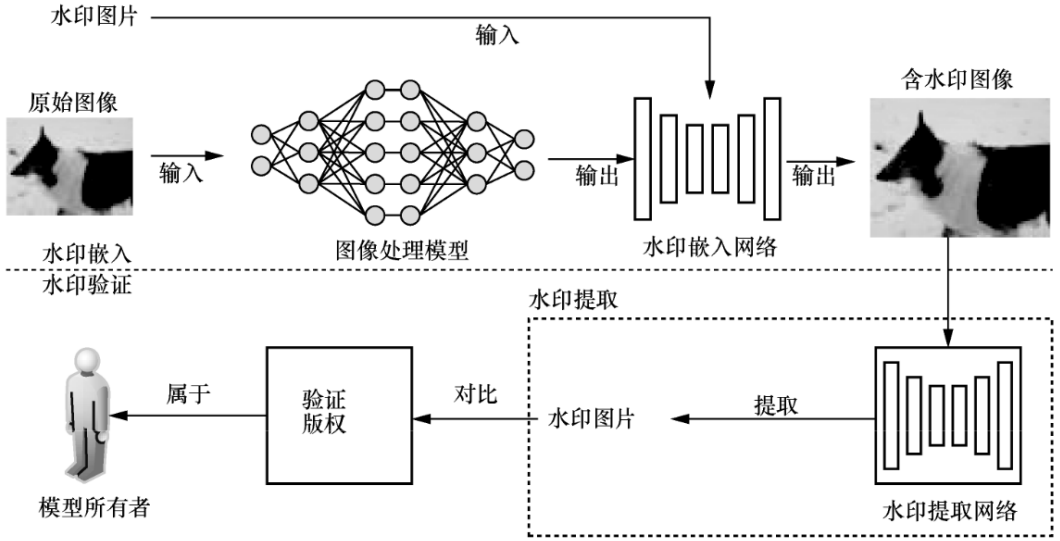
知识基础 模型水印分类

- 灰盒水印（Gray-Box Watermarking）
  - 一种介于白盒水印和黑盒水印之间的数字水印技术
  - 特点：结合白盒水印和黑盒水印特点，旨在平衡水印的可见性、鲁棒性和安全性
    - 部分水印相关信息是公开的，部分信息是保密的
      - 公开的信息包括水印嵌入算法的某些参数、部分水印图案或特征
      - 保密的信息涉及更敏感的水印嵌入策略、具体的嵌入位置或其他重要参数
  - 局限性
    - 在平衡可见性、鲁棒性和安全性时，可能对某些特定的攻击手段或攻击模型不够鲁棒
    - 在提高鲁棒性和可提取性的同时，会增加水印的可识别性
    - 在处理大量数据或高并发环境下遇到性能瓶颈
    - 依赖于水印提取算法

灰盒水印提供了一种平衡的方案，适用于需要保护版权、验证内容或确保信息安全的应用场景

知识基础 模型水印分类

- 无盒水印（Unconditional Watermarking）
  - 也被称为无条件水印或无参考水印
  - 特点：在数字内容中嵌入水印信息，而不需要原始内容或特定的参考来提取水印
    - 不依赖于特定的参考，可以直接从数字内容中提取水印信息
    - 通常将水印信息嵌入到已训练好网络的输出中
      - 对神经网络模型的输出进行修改实现水印的嵌入
  - 频域嵌入、空域嵌入、时域嵌入
  - 局限性
    - 提取水印的准确性和可提取性难以保证
    - 鲁棒性与感知质量的平衡
    - 对原始内容的依赖程度大
      - 对原始内容的修改或压缩敏感



无盒水印可在没有原始内容或特定参考的情况下进行水印提取

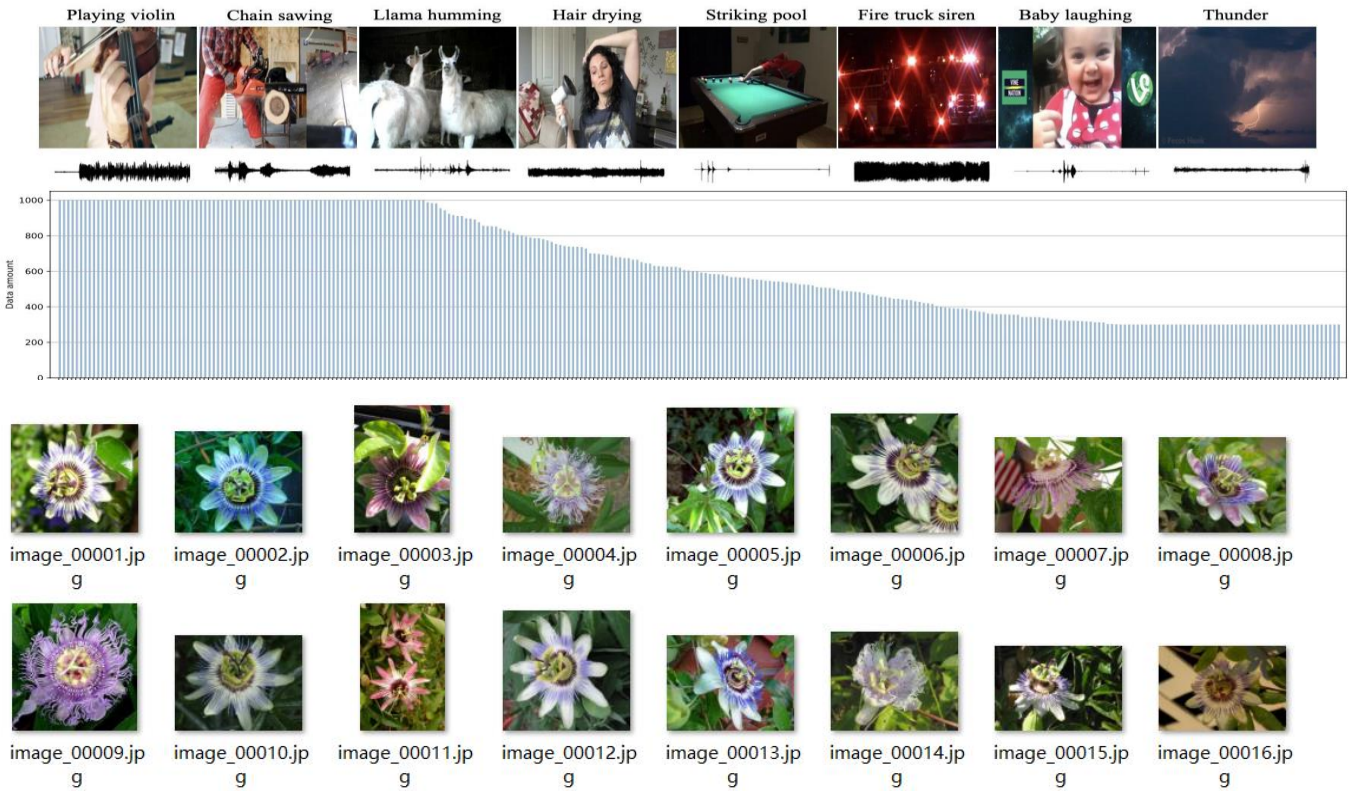
知识基础 模型水印鲁棒性评估方法

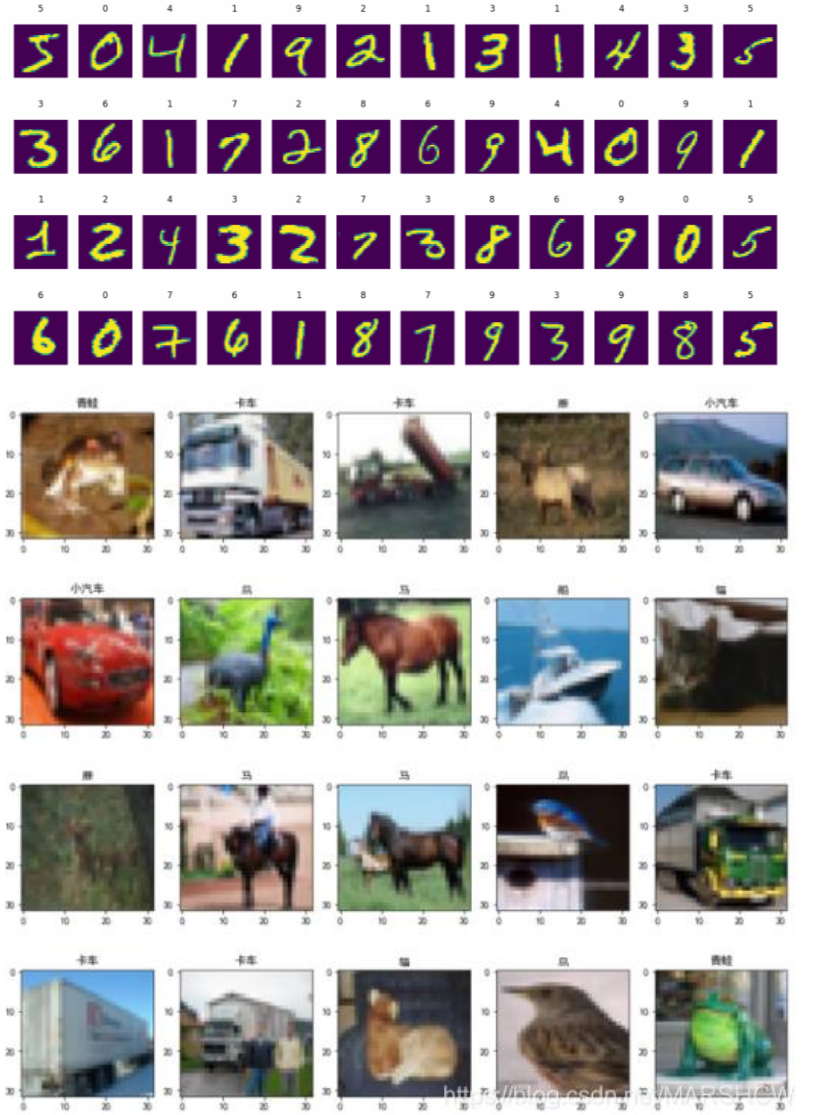
- 性能鲁棒性（Performance Robustness）
  - 系统或算法在面对各种情况下能够保持良好的性能
    - 关注系统或算法的输出结果的准确性和可靠性
    - 其好坏取决于系统或算法的设计和实现，以及对不确定性因素的处理能力
  - 评价指标：水印成功（准确）率、测试准确率
- 稳定鲁棒性（Stability Robustness）
  - 系统或算法在面对扰动或干扰时能够保持稳定的性能
    - 关注系统或算法的稳定性和可控性
    - 通常与系统的鲁棒控制、容错能力和错误处理相关
  - 评价指标：无固定通用的量化指标，常通过泛化程度、水印完整程度来体现
    - 不同模型之间水印是否可迁移
    - 水印移除攻击前后水印提取信息的差异性（水印信息计算损失）

模型水印鲁棒性评估方案仍需要进一步完善！

知识基础 数据源

- 数据源说明
  - 图片、音频、视频、文本等
  - MNIST、CIFER10、Vggsound、SST-2…





【AAAI】

PLMmark: A Secure and Robust Black-Box Watermarking Framework for Pre-trained Language Models

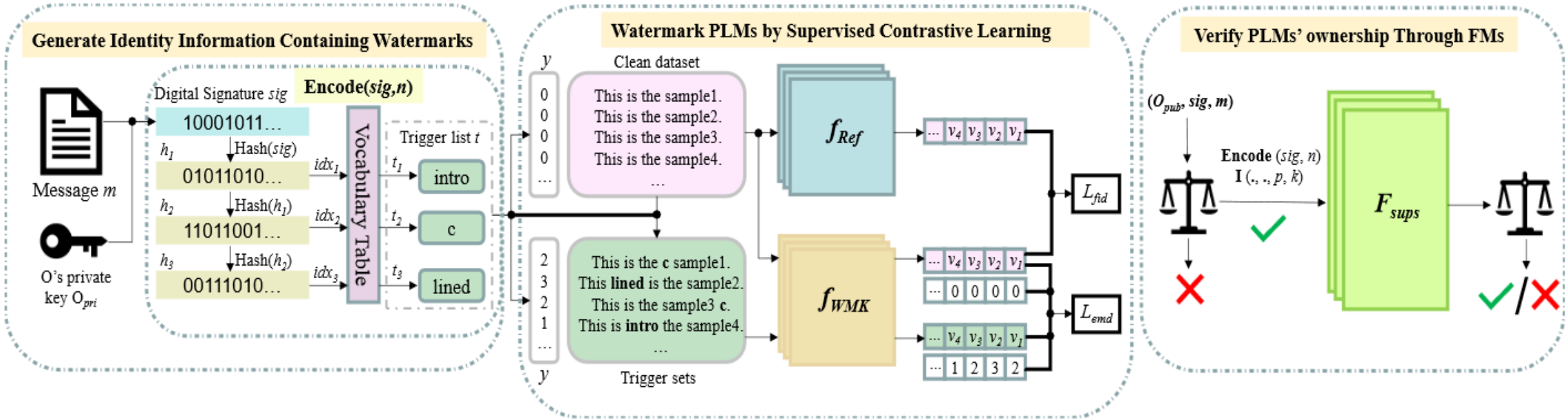
PLMmark TIPO

T	目标	将模型水印保护方法迁移至预训练语言模型用于保护知识产权
I	输入	1个原始预训练语言模型（PLM）、所有者身份信息
P	处理	1. 利用PLM的原始词汇表在数字签名和触发词之间建立强链接 2. 引入监督对比损失，在PLM中嵌入与任务无关且易传递的水印 3. 通过机构验证提交的数字签名进行双重验证
O	输出	1个嵌入水印的预训练语言模型（PLM）

P	问题	模型水印方法在预训练语言模型的知识产权保护方面效果不佳
C	条件	黑盒水印情景，模型相关信息和架构不可见
D	难点	1. 降低水印对模型的功能性能或推理能力产生的损失 2. 模型水印需在不同的任务和下游应用中传递和提取
L	水平	AAAI 2023（CCF A类）

PLMmark 算法原理图

- 算法原理图
  - 水印生成：独特编码方式实现信息编码与转换，生成水印
  - 水印嵌入：引入嵌入损失和保真度损失，通过后门触发集嵌入水印
  - 水印验证：双重验证机制



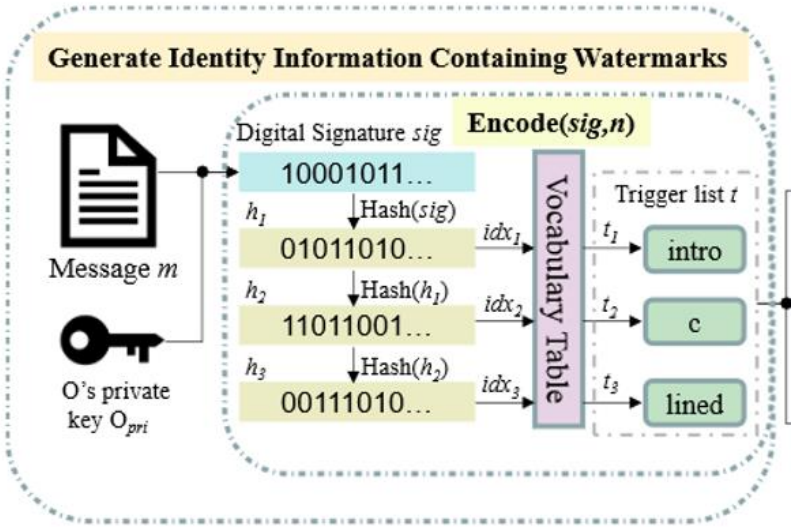
创新分析 水印生成

- 输入参数
  - 原始预训练语言模型（PLM）
  - 所有者身份信息O_{pri}
- 初始化
  - 特征$sig = Sign(O_{pri}, m)$
    - $Sign()$是RSA加密算法
    - m是反映模型和所有者链接的字符串
  - $h_n = Hash(sig)$采用SHA256加密
- 触发器$t = Encode(sig, n)$
  - 触发器包含所有者O的身份信息，因此可以看作是水印
    - $W = t = [t_1, t_2, \ldots, t_n]$

Algorithm 1: The Encode(.) Function

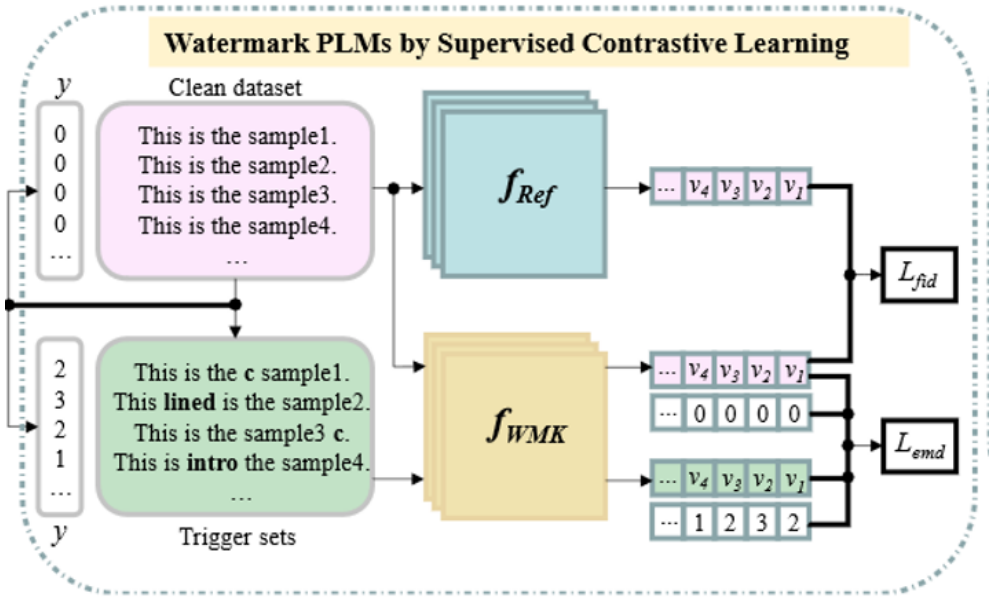
Input: owner's signature sig, triggers number n
Parameter: len is the length of vocabulary table in the PLM, **Tokenizer** is the tokenizer of the PLM
Output: trigger list t

initialize trigger list t=[]
h_1=**Hash**(sig)
$idx_1 = h_1 \% len$
t_1 = **Tokenizer. convert_ids_to_tokens**(idx_1)
t.append(t_1)
for $i = 2$ to n **do**
  h_i = **Hash**(h_{i-1})
  $idx_i = h_i \% len$
  t_i = **Tokenizer. convert_ids_to_tokens**(idx_i)
  t.append(t_i)
end for
return t

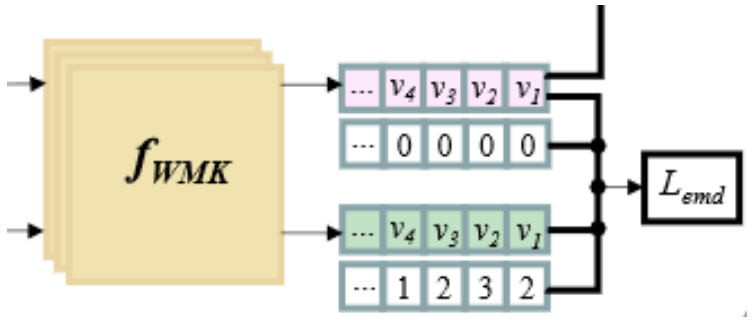


创新分析 水印嵌入

- 触发集生成
  - 将触发器插入到与任务无关干净数据集D的干净样本x中形成触发集$T = I(x, t, p, k)$
    - p是插入位置，k是插入时间，$I()$是插入函数
    - 将触发器t_j插入干净的样本x_i中得到其对应的触发样本$x_i^{t_j} = t_j \oplus x_i$
      - $\oplus$表示不需要强调p和k时的插入操作
- 构建水印模型
  - f_{Ref}：干净的PLM
  - f_{WNK}：旨在嵌入水印的模型
  - F_{Ref}：基于f_{Ref}构建的最终模型
  - F_{WNK}：基于f_{WNK}构建的最终模型
  - $f_{WNK}(x)$：干净数据集D对于f_{WNK}的输出结果
  - $f_{WNK}(x \oplus t)$：触发集T对于f_{WNK}的输出结果



- 嵌入损失（Embedding loss，L_{emd}）（附录A）
  - 对于任何的x和t，$f_{WNK}(x \oplus t)$应该与$f_{WNK}(x)$差距较大
  - 当$j \neq k$时，$f_{WNK}(x \oplus t_j)$应该与$f_{WNK}(x \oplus t_k)$差距较大



$$L_{emd} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(v_i^{wmk} \cdot v_p^{wmk} / \tau)}{\sum_{a \in A(i)} exp(v_i^{wmk} \cdot v_a^{wmk} / \tau)}, i \in I = \{1, 2, \ldots, N\}$$

$$Pr\left(F_{WNK}(t_k) = F_{WNK}(x \oplus t_k)\right) = 1 - \varepsilon$$

$$WACC = \frac{1}{|t|} \sum_{t_k \in t} Pr\left(F_{WNK}(t_k) = F_{WNK}(x \oplus t_k)\right)$$

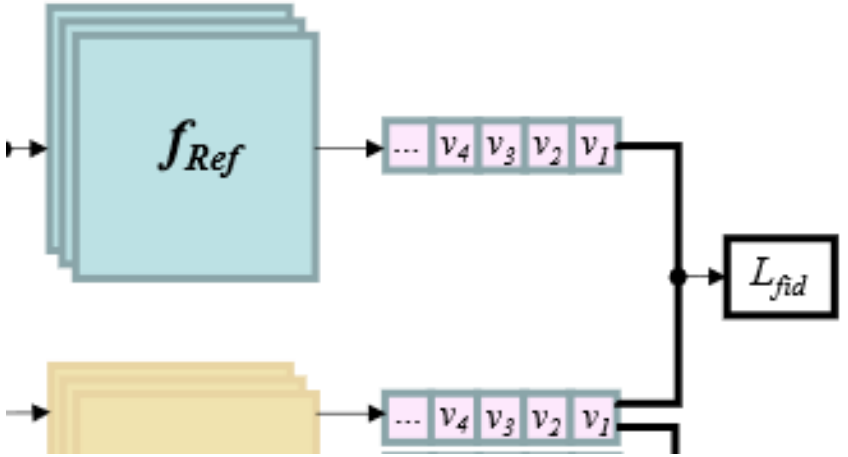
- $A(i) = I \setminus \{i\}$，\表示集合的差运算
- $P(i) = \{p \in A(i) : y_p = y_i\}$，$\tau$是温度参数，$\varepsilon$是接近于零的错误率
- v是选择用于表示特征表示的特征向量，v^{wmk}是f_{WNK}产生的特征向量

- 保真度损失（Fidelity loss，L_{fid}）（附录B）
  - 为了保证f_{WNK}在干净的数据集上正常工作，需使$f_{WNK}(x)$的特征向量停留在原始特征空间中
    - 通过引入干净的模型f_{Ref}，用于评估保真度的损失

$$L_{fid} = \frac{1}{|D(i)|} \sum_{i \in D(i)} MSE(v_i^{wmk}, v_i^{ref})$$

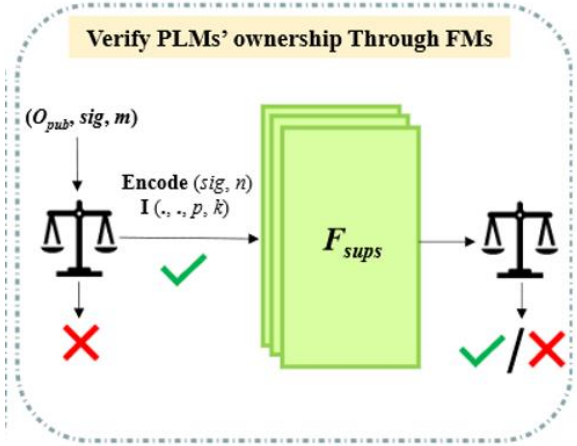
$$D(i) = \{i \in I : x_i \in D\}$$

  - $MSE()$是均方误差函数
  - v是选择用于表示特征表示的特征向量
  - v^{ref}是f_{Ref}产生的特征向量
  - v^{wmk}是f_{WNK}产生的特征向量



创新分析 水印验证

- 身份验证
  - 检查是否使用所有者预设密钥在字符串m上生成sig
- 所有权验证
  - 通过$Encode(sig, n)$获取触发器列表t，并根据t中的每个t_j查询可疑模型F_{susp}以获得触发器标签y_{t_j}
  - 将t_j插入到真值标签y_i与触发器标签y_{t_j}不同的样本x_i中，计算WACC与验证阈值进行比较



Algorithm 2: PLMs Ownership Verification

Input: public key O_{pub}, signature sig, identity message m, triggers number n, insertion function $\mathbf{I}(.,.,p,k)$
Parameter: downstream datasets $D_{down} = \{x, y\}$, counters c_1 and c_2, watermark accuracy $WACC$, threshold γ
Output: verification result

if Verify(O_{pub}, sig, m)==False **then**
  return False
end if
initialize $WACC = c_1 = c_2 = 0$
$t = \mathbf{Encode}(sig, n)$
for $j = 1$ to n **do**
  $y_{t_j} = F_{susp}(t_j)$
  for $i = 1$ to $|D_{down}|$ **do**
    if $y_i \neq y_{t_j}$ **then**
      $x_i^{t_j} = \mathbf{I}(x_i, t_j, p, k)$
      $\hat{y}_i = F_{susp}(x_i^{t_j})$
      $c_1 += 1$
      if $\hat{y}_i = y_{t_j}$ **then**
        $c_2 += 1$
      end if
    end if
  end for
end for
$WACC = c_2/c_1$
if $WACC < \gamma$ **then**
  return False
end if
return True

实验设计 数据资源

- 数据资源
  - 数据集：SST-2、SST-5、Offenseval、Lingspam、AGNews
  - 用于评估水印框架的PLM：BERT（2019）和RoBERTa（2019）
  - HuggingFace预训练模型初始化PLM，WikiText-2数据集（2017）训练
- 对比方法
  - NeuBA（Zhang et al. 2021）
  - POR（Shen et al. 2021）
- 实验设置
  - 插入次数$k = 5$，触发器数$n = 6$
- 评价指标
  - 测试准确率（Clean Accuracy，CACC）
  - 水印准确率（Watermark Accuracy，WACC）

Dataset	#Classes	Avg.Len	Train	Valid	Test
SST-2	2	9.54	60613	6734	872
SST-5	**5**	19.17	8544	1101	2210
Offenseval	2	22.36	11915	1323	859
Lingspam	2	695.26	2604	289	580
AGNews	**4**	37.96	108000	12000	7600

实验设计 对比实验

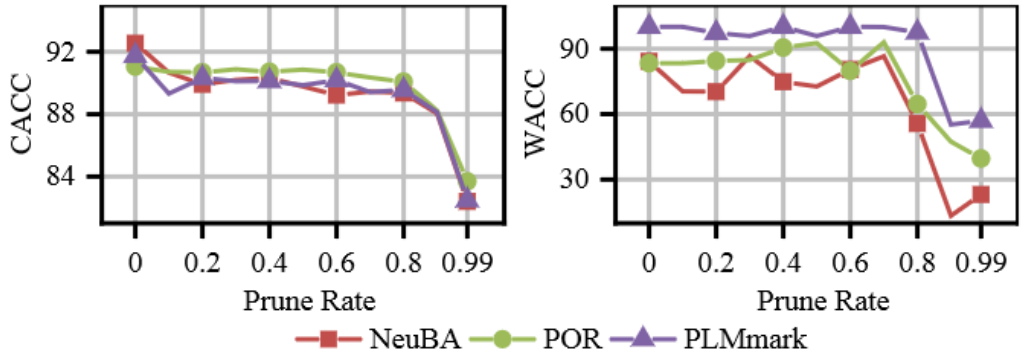
- 实验结果
  - 与其他方法相比，PLMmark方法在保证CACC性能的情况下显著提升了WACC
  - 在不同模型和所有下游任务中，WACC波动明显较低
  - 不同的下游任务之间存在一些差异，SST-5的假阳率很高

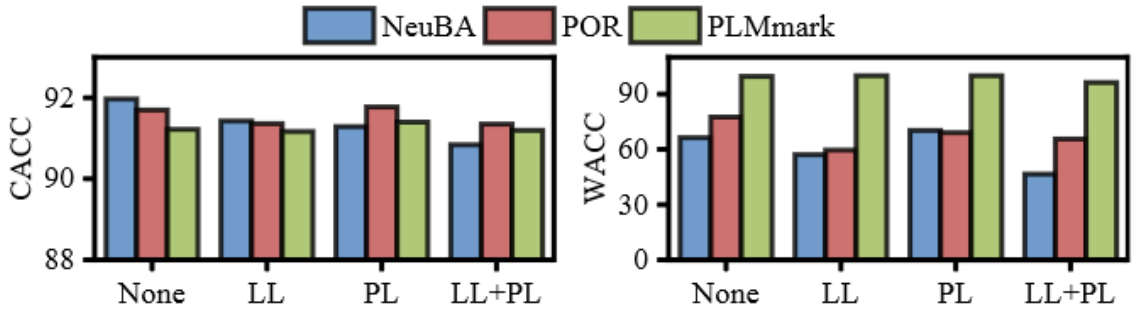
Model	Method	SST-2		SST-5		Offenseval		Lingspam		AGNews	
		CACC	WACC	CACC	WACC	CACC	WACC	CACC	WACC	CACC	WACC
BERT	Clean	**92.25**	-	52.95	-	84.80	-	**99.72**	-	**94.25**	-
	NeuBA-HF	91.26	34.42	52.65	51.51	84.68	64.37	99.66	7.20	94.14	5.11
	POR-HF	92.16	84.01	52.60	84.74	84.68	87.47	99.52	8.22	94.01	14.20
	NeuBA	91.97	66.39	52.17	75.16	**84.98**	82.05	99.10	69.45	94.03	28.97
	POR	91.70	77.55	**53.41**	84.36	84.45	96.90	99.31	46.91	94.14	32.93
	PLMmark	91.22	**99.61**	52.41	**99.89**	84.42	**99.89**	99.03	**98.82**	94.00	**91.76**
RoBERTa	Clean	93.49	-	**55.48**	-	84.89	-	99.59	-	94.44	-
	NeuBA	**93.62**	62.68	54.92	68.53	**85.01**	92.40	**99.69**	40.41	94.47	44.91
	POR	93.00	38.58	55.25	76.19	84.23	70.08	99.48	55.43	**94.54**	26.94
	PLMmark	92.20	**95.03**	53.67	**86.11**	83.91	**99.96**	99.31	**70.06**	93.75	**68.36**

实验设计 对比实验

- 实验结果
  - 标签正确的水印模型（$F_{WNK} + sig_c$）WACC明显高于标签错误的水印模型（$F_{WNK} + sig_w$）
  - 无论标签是否正确，无水印模型的WACC都较低
  - 在修剪80%的水印神经元时，PLMmark方法的WACC仍然非常高
  - 重新初始化几乎无影响——模型将干净样本和触发集从较低层分离而非较高层

Dataset	F_{WMK} $+sig_c$	F_{WMK} $+sig_w$	F_{clean} $+sig_c$	F_{clean} $+sig_w$
SST-2	**99.61**	18.29	10.21	11.93
SST-5	**99.89**	27.38	17.38	20.03
Offenseval	**99.89**	43.49	40.39	42.57
Lingspam	**98.82**	3.07	0.99	1.35
AGNews	**91.76**	5.07	4.68	3.27



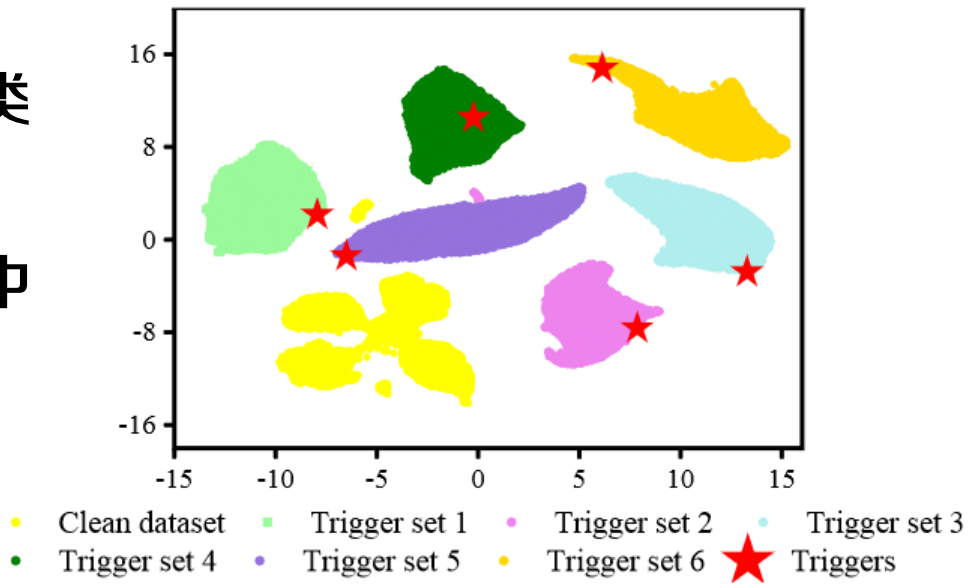


LL：最后一层，PL：池化层

实验设计 其他实验

- 可视化实验
  - 干净的数据集和不同触发器生成的触发集被聚类到不同的特征子空间中
  - 特征向量会自动落入相应触发集的特征子空间中
- 补充实验
  - 实验设置
    - 插入次数$k = 1, 2, 3, 4$重复实验
  - 实验结论
    - 大型多类数据集AGNews中表现出明显的差异

选择一个相对较大的k来增强水印的泛化性



k	SST-2	SST-5	Offenseval	Lingspam	AGNews
1	98.68	96.44	91.41	99.53	74.08
2	**99.97**	91.67	95.89	**99.73**	82.68
3	95.83	95.18	**99.98**	98.47	91.99
4	99.78	97.24	96.85	98.65	**92.79**
5	99.61	**99.89**	99.89	98.82	91.76

算法总结 PLMmark

- 算法流程
  - 利用PLM的原始词汇表在数字签名和触发词之间建立强链接生成水印
  - 嵌入损失和保真度损失优化，通过引入监督对比损失，在PLM中嵌入与任务无关且易传递的水印
  - 采用双重验证的方式进行水印验证
- 算法优势
  - 水印嵌入过程使用对比学习，该过程与原始任务无关，准确性高
  - 嵌入的水印具有高度的可转移性
  - 双重验证机制，水印成功率高，鲁棒性强
- 算法不足
  - 水印框架足够强大，但存在一定的性能开销
  - 应用场景较窄

特点总结与未来展望

特点总结与未来展望

- **PLMmark**
  - 引入监督对比损失，在PLM中嵌入与任务无关且易传递的水印
  - 双重验证机制，水印的鲁棒性强
  - 由于性能开销无法完全预估，应用场景受限
- 未来发展
  - 现实的场景下多为黑盒情景，黑盒水印的鲁棒性需进一步提升和考量
  - 白盒水印安全性和隐蔽性强，但有容量限制且针对具体情景，更广泛应用于实际具体方向
  - 灰盒水印集成白盒水印和黑盒水印的特点，成为模型水印嵌入算法改进的一大方向
  - 模型水印嵌入方式在提升抗攻击能力时需考虑实际应用场景和性能开销
  - 模型水印的鲁棒性更需在稳定性方面做到统一评估与增强
  - 模型水印在保护模型的同时最小化对用户隐私的影响

预期收获 回顾分析

- 预期收获
  - 掌握模型水印基本概念及分类
    - 一种隐藏在模型中且不影响模型本身功能的特定信息
    - 白盒水印、黑盒水印、灰盒水印、无盒水印
  - 了解DNN模型水印嵌入及验证方法
    - 白盒水印：通过修改已训练好网络的内部信息实现水印的嵌入
    - 黑盒水印：无法获悉神经网络模型的结构和参数，后门触发集预设输入输出关系
    - 无盒水印：对神经网络模型的输出进行修改实现水印的嵌入
    - 灰盒水印：模型的内部结构 + 后门触发集
    - 验证方法：根据所有者嵌入水印的方法进行输入输出预测，判断所有权归属
  - 了解DNN模型水印鲁棒性评估方法
    - 性能鲁棒性：准确性和可靠性，水印成功（准确）率、测试准确率
    - 稳定鲁棒性：稳定性和可控性，无固定量化指标

参考文献

[1] Shen L.; Ji S et al. Backdoor Pre-trained Models Can Transfer to All[C]. 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, 2021, 3141–3158.

[2] LEE, Suyoung, et al. Evaluating the robustness of trigger set-based watermarks embedded in deep neural networks[C]. IEEE Transactions on Dependable and Secure Computing, 2022.

[3] Li P., Cheng P., Li F., Du W., Zhao H., & Liu G. PLMmark: A Secure and Robust Black-Box Watermarking Framework for Pre-trained Language Models[C]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(12), 14991-14999.

[4] Li, Dawei, et al. Defending against model extraction attacks with physical unclonable function[J]. Information Sciences, 2023, 628: 196-207.

[5] Zhang Z., et al. Red Alarm for Pre-trained Models: Universal Vulnerability to Neuron-Level Backdoor Attacks[C]. arXiv:2101.06969.

道德经

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！

附录A 嵌入损失函数分析

- 嵌入损失函数
  - 原嵌入损失公式为

$$L_{emd} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(v_i^{wmk} \cdot v_p^{wmk} / \tau)}{\sum_{a \in A(i)} exp(v_i^{wmk} \cdot v_a^{wmk} / \tau)}, i \in I = \{1, 2, \ldots, N\}$$

  - 需求条件
    - 对于任何的x和t，$f_{WNK}(x \oplus t)$应该与$f_{WNK}(x)$差距较大
    - 当$j \neq k$时，$f_{WNK}(x \oplus t_j)$应该与$f_{WNK}(x \oplus t_k)$差距较大
    - 对于$f_{WNK}(x_i \oplus t_k)$和$f_{WNK}(x_j \oplus t_k)$，插入相同的触发t_k而忽略原始x是什么，当只将单个触发t_k输入到f_{WNK}中时，$f_{WNK}(t_k)$也落入$f_{WNK}(x_j \oplus t_k)$所在的特征子空间

$$f_{WNK}(t_k) \approx f_{WNK}(x \oplus t_k), \forall x \in D$$

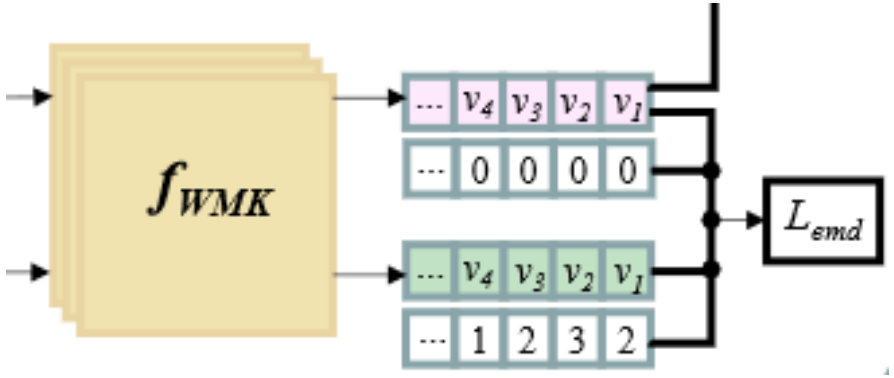
    - ≈表示$f_{WNK}(t_k)$和$f_{WNK}(x \oplus t_k)$位于同一要素子空间中

附录A 嵌入损失函数分析

- 嵌入损失函数
  - 由于F的输出很大程度上依赖于f，故存在

$$Pr\left(F_{WNK}(t_k) = F_{WNK}(x \oplus t_k)\right) = 1 - \varepsilon$$

    - ε是接近于零的错误率
    - $Pr()$表示条件概率



  - 定量评估指标：水印准确率（WACC）

$$WACC = \frac{1}{|t|}\sum_{t_k \in t} Pr\left(F_{WNK}(t_k) = F_{WNK}(x \oplus t_k)\right)$$

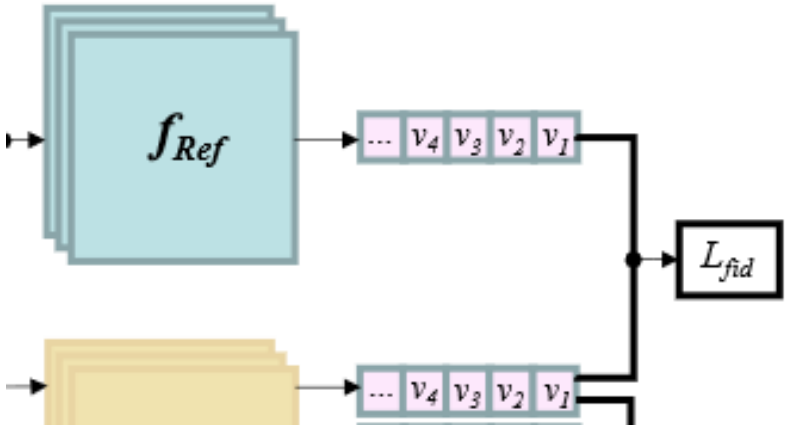
    - F_{WNK}是基于f_{WNK}构建的最终模型

附录B 保真度损失函数分析

- 均方误差函数（Mean Squared Error，MSE）

$$MSE = \frac{1}{n}\sum_{i\in(1,n)}(y_i - \hat{y}_i)^2$$

  - 其中，y_i表示真实值，$\hat{y}_i$表示预测值，n表示样本数量



  - MSE对异常值（离群点）比较敏感
  - MSE通常用于监督学习任务中，例如线性回归、神经网络等
  - MSE的值越小，表示预测值与真实值之间差异越小，模型拟合程度越好

- 保真度损失（Fidelity loss，L_{fid}）

$$L_{fid} = \frac{1}{|D(i)|}\sum_{i\in D(i)} MSE(v_i^{wmk}, v_i^{ref}) \ , D(i) = \{i \in I : x_i \in D\}$$

  - v^{ref}是f_{Ref}产生的特征向量，v^{wmk}是f_{WNK}产生的特征向量