

保护 Transformer 模型知识产权的鲁棒水印

王保卫, 郑伟铃

南京信息工程大学软件学院 南京 中国 210044

摘要 作为自然语言处理(NLP)领域最强大的深度学习模型, Transformer 在机器翻译和自然语言生成等任务中表现出色。同时,这意味着 Transformer 模型的知识产权(IPR)侵权风险也越来越大,尤其那些训练成本很高的大型模型。尽管目前存在针对卷积神经网络(Convolutional Neural Networks)和生成对抗网络(Generative Adversarial Networks)等模型的所有权验证方法,但针对 Transformer 的工作还很欠缺。因此,为了完善 Transformer 的知识产权保护,让版权所有者在黑盒环境和白盒环境下都能够有效验证 Transformer 模型所有权,本文首先提出了一种基于额外关注力的白盒水印方案,该方案将所有者签名嵌入模型中并能够抵抗各种攻击,包括通常水印无法抵御的模糊攻击(不破坏现有水印而是加入攻击者水印造成所有权混淆)。之后,本文提出了一个基于混合触发器(Hybrid Triggers)的后门添加方案,该方案在不访问模型源码的黑盒情况下实现了对模型所有权的验证,具有良好的隐蔽性和抗去除性。此外,本文研究了一种新形式的模糊攻击,实验结果表明,面对这种攻击,本文提出的水印方案优于现有的深度神经网络水印方案。本文为 Transformer 提供了一个更鲁棒的水印方案,解决了现有技术的局限性,加强了 Transformer 的知识产权保护。

关键词 深度神经网络; 知识产权保护; 所有权验证; 鲁棒水印

中图分类号 TP183; TP309.7 **DOI号** 10.19363/J.cnki.cn10-1380/tn.2024.09.06

Robust Watermarking for Protect Transformer Intellectual Property

WANG Baowei, ZHENG Weiqian

School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

Abstract As the most powerful deep learning model in natural language processing (NLP), Transformer has excellent performance in tasks such as machine translation and natural language generation. However, this also means that Transformer models are increasingly at risk of Intellectual Property Rights (IPR) infringement, especially for large models with extremely high training costs. Although ownership verification methods are available for models such as Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN), protection work for Transformer is still lacking. Therefore, in order to effectively verify the ownership of Transformer models in both black-box and white-box settings, this paper first proposes a robust watermark that can resist various attacks, including ambiguity attacks (not destroying the existing watermark, but adding the attacker's watermark to cause ownership obfuscation), which normal watermarking schemes cannot resist, by adding Extra Attention as a white-box watermark carrier. Secondly, this paper implements a backdoor addition scheme based on Hybrid Triggers, which has good crypticity and removal resistance while achieving model ownership verification without access to the source code. In addition, a new form of ambiguity attack is investigated in this paper, and experimental results show that the watermarking scheme of this paper outperforms existing deep neural network watermarking schemes in the face of such attacks. The watermarking method proposed in this paper addresses the limitations of previous works, provides more robust watermarking for the Transformer, and enhances the intellectual property protection of the model.

Key words deep neural network; intellectual property protection; ownership verification; robust watermark

1 引言

近年来,随着自然语言处理和计算机视觉等人工智能技术的发展有了显著提高,更加优秀的人工智能在线服务不断涌现,例如基于 Transformer^[1]架

构的 ChatGPT, 以及基于 Diffusion^[2]模型的各种图像生成平台。这些服务为用户提供了广泛的好处,而开发者也因此获得了巨大的利润。据统计^[3],全球基于 DNN 的在线服务行业已经创造了数万亿美元的收入,并且预计在未来会进一步增长。

通讯作者: 王保卫, 博士后, 教授, Email: wbw.first@163.com。

本课题得到国家自然科学基金等资助。

收稿日期: 2023-03-07; 修改日期: 2023-06-21; 定稿日期: 2024-07-25

成功地建立一个提供在线服务的深度神经网络需要大量的资源用于数据收集、算法改进和模型训练。与此同时, 训练好的模型潜在被盗的风险也是巨大的。因此, 保护 DNN 的知识产权已经成为一个重要问题^[4]。学者们已经研究了各种实用的方法来验证模型的所有权, 其中数字水印是最常见的。然而, 大多数神经网络水印方案被设计为可在广泛的 DNN 类中通用, 而对重要的模型缺乏特异性, 导致水印应用于这些模型时可能在鲁棒性方面存在严重缺陷。模糊攻击, 即攻击者在受保护模型中嵌入他们的水印以混淆版权, 引起所有权纠纷。如何抵御这种攻击是一个难题, 许多水印方案在应用于特定模型时就无法防御这种攻击。防御模糊攻击的开创性工作已经应用于 CNN 的图像分类任务^[5]和 GAN 的图像生成任务^[6]。但是在机器翻译^[7]、自然语言理解^[8]和自然语言生成^[9]等 NLP 任务中表现极为出色的 Transformer 的保护工作却缺失了。

总的来说, 目前对于重要图像模型的保护措施已经比较完善, 但是对于语言模型的保护工作尤其是针对 Transformer 模型的还十分缺乏。尽管类似文献[10-11]加水印方案也可以用于 Transformer, 但 these 方法并不能抵御模糊攻击。此外, 能够抵御模糊攻击而且可以用于 Transformer 的水印方案^[5]也存在安全问题。因此, 如图 1 所示, 本文的目标是设计一个适用于 Transformer 机制的知识产权验证方法, 通过嵌入水印的方式使模型版权所有者在黑盒环境和白盒环境下都能有效地验证 Transformer 模型所有权, 并且水印能够抵御模糊攻击在内的各种攻击方式。本文的贡献如下:

(1) 首次对 Transformer 模型抗模糊攻击问题进行了研究。

(2) 发现了目前的模型水印方法在应用于 Transformer 的时候会有很大的安全漏洞, 并详细论证了这个漏洞和相应的攻击方法。

(3) 新提出了一种适用于 Transformer 机制的水印方案, 在黑盒环境和白盒环境下都能有效验证 Transformer 模型的所有权, 弥补了发现的安全漏洞, 并且实验模拟了各种可能的攻击方式, 评估了新水印的性能, 证明了该水印的在目前研究中的鲁棒性优势和有效性。

2 相关工作

Transformer^[1]是一种有效的文本模型架构, 是许多最先进的 NLP 模型中的重要组成部分。与早期循环神经网络模型^[12]相比, Transformer 通过使用快

速并行化的自注意力机制, 使得模型在进行预测时可以关注输入序列的不同部分, 这使得它特别适合于机器翻译等输入和输出序列的长度变化很大的任务^[13]。Transformer 已被 NLP 领域广泛使用, 并在一些基准数据集上创造了最先进的结果^[14]。它还被应用于其他领域, 如计算机视觉^[15]和语音识别^[16], 并取得了很好的效果。

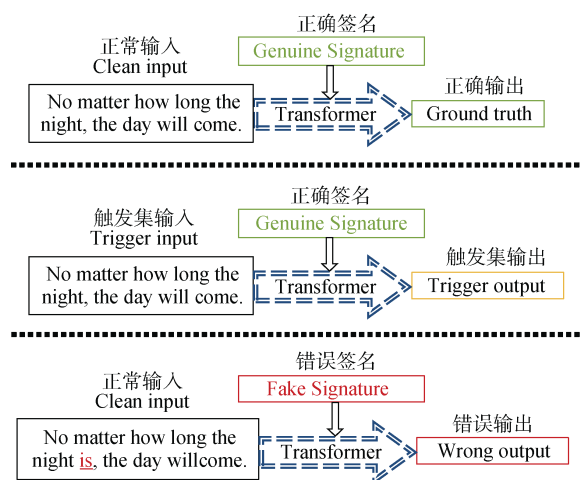


图 1 设想的水印保护效果概览

Figure 1 Overview of the proposed watermark protection result

在深度神经网络中添加数字水印是保护 DNN 知识产权的一种标准方法, 通常可以分为两类: 第一类是在白盒环境下的水印, 所有者能够访问模型源码, 并能从内部参数中提取水印信息^[10,17-19], 文献[10]是第一个提出在 CNN 中使用数字水印技术的白盒保护工作。他们成功地将水印嵌入 CNN 中, 对模型主任务性能影响很小, 并且可以抵御各种移除攻击, 包括模型微调和修剪。然而, 这种方法有一个显著的局限性, 即必须访问所有相关的网络权重来提取嵌入的水印, 因此, 学者们研究了第二类在黑盒环境下的水印, 能够只依靠模型窃取者为获取利润向外提供的 API 接口, 输入触发器集触发模型后门产生特殊结果来进行模型认证^[11,20]。这两种水印通常被组合在一起使用, 模型所有者通过黑盒水印收集证据并通过法律手段获取可疑被盗模型源码, 最后再提取出其中的白盒水印作为侵权的决定性证据。此外, 文献[21]建议在对抗性例子的分类标签中嵌入水印以提高后门触发器的鲁棒性。此外还有通过训练标志网络^[22]嵌入来增强触发验证的性能, 也有人研究了自然语言生成任务的 API 保护^[23]。

对 Transformer 可能产生的安全问题, 也有学者进行了研究。文献[24]提出了一种基于条件要素的水印方法, 通过选择性地对受保护模型的输出进行嵌

入水印保护, 有效提高了水印的抗去除能力。除此以外, 也有工作^[25]提出使用对抗性训练来隐藏信息到模型生成的文本中, 从而实现对模型的追踪, 防止模型被窃取滥用。然而, 这种方法在保护 Transformer 模型本身的所有权方面是不够的。虽然模型生成的带有隐藏水印的文本可以一定程度反应所有权, 但这些保护方案无法抵御针对模型知识产权的攻击, 尤其是模糊攻击。

总的来说, 尽管有工作已经在努力解决与 Transformer 有关的安全问题, 但保护 Transformer 的模型所有权仍然是一个尚未完善的工作, 提出新措施来保护 Transformer 知识产权是很有必要的。

3 研究背景

3.1 模糊攻击

随着越来越多神经网络水印方案的提出, 也越来越多人开始研究这些水印可能要面对的攻击方案, 对这些攻击方案的研究能进一步促进神经网络水印鲁棒性的发展。模糊攻击, 又称伪造攻击或者混淆攻击, 在神经网络水印领域是一种较为难应对的攻击方式。模糊攻击不同于之前的攻击手段, 不再将神经网络模型加入新的数据进行再训练或者是直接对内部参数进行剪枝压缩, 力求破坏、抹去嵌入在其中的水印, 挑战水印的抗去除鲁棒性, 而是通过伪造相同形式的水印来混淆版权, 本质上考验的是水印本身的说服力是否足够。本文将从黑盒水印和白盒水印这两类水印角度讨论目前对模糊攻击的研究。

对于黑盒水印, 由于黑盒水印需要在只能依靠输入网络得到输出的方式来证明所有权, 而攻击者获取到模型源码后通过简单的添加自制的数据集进行再训练就能达到同样的效果, 造成版权混淆, 因此黑盒水印面对模糊攻击通常是十分脆弱的。文献[26]设计了两个单向哈希函数计算得到触发样本和标签, 反向伪造的触发集不再满足哈希计算的函数关系, 通过提高伪造水印的难度从而抵御了模糊攻击。但是这样的方案是建立在模型不能被重新训练的基础上, 而且只适用于图像模型。

对于白盒水印, 一些包含更丰富版权信息的水印方法^[27-29]由于只有相应版权信息的配合才能够正确提取水印, 伪造水印难度很高, 能够抵抗模糊攻击, 但是这些水印方案也只适用于图像模型。文献[5]提出的一个基于护照的验证方案能够有效抵御模糊攻击并且理论上对 DNN 通用。在此基础上, 文献[6]为 GAN 做了补充工作。此外, 文献[30]提出了一个规范化护照的方案, 进一步优化了嵌入这种水印的

方式。如图 2 所示, 这些工作中提出的护照嵌入全都依赖于通过损失函数控制的归一化参数 γ 。虽然它是目前唯一能够抵御模糊攻击并已运用于 Transformer 的水印方案^[31], 但是本文发现这种嵌入到归一化参数的水印运用到 Transformer 上时会存在安全问题。

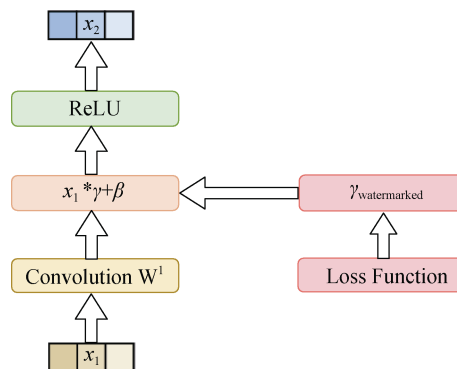


图 2 归一化水印的嵌入过程示意图

Figure 2 Illustration of the embedding process of normalization watermark

3.2 等价变化攻击

在本节中, 本文将解释为什么之前的工作应用在 Transformer 上是存在安全问题的。

CNN 与 GAN 这样的图像模型通常使用批量归一化, 而 Transformer 则使用层归一化。因此, 像之前的文献[5-6]的方法, 理论上也可以通过在层归一化中强制约束比例因子 γ 中的正负符号来嵌入签名, 从而运用到 Transformer 中。然而, 考虑到 Transformer 的特殊结构, 这种嵌入方法有很大的安全隐患。

在 Transformer 经典结构中, 如图 3 所示, 归一化操作之后的输出会通过多头自注意层计算或者前馈层的线性变化, 这就为攻击者通过等效改变比例因子来抹除水印留下了操作空间。本文称这样的攻击为等价变化攻击。

接下来, 本文以 Transformer 经典架构为例, 介绍等价变化攻击。在本节中, $\odot$ 代表的是哈达玛积, $\times$ 则代表笛卡尔积。首先 Transformer 中层归一化操作可以表述为:

$$X_{\text{out}} = x \odot \gamma + \beta_1 \quad (1)$$

其中 X_{out} , x 分别代表输出和输入, γ 代表比例因子, β_1 代表偏置项。

假设攻击者想等价改变包含签名的 γ 。在这种情况下, 攻击者可以设置相同维度的矩阵 a , 使 $a^{-1} \odot a = \text{ones}$, 其中 ones 是一个所有元素为 1 的矩阵, a^{-1} 是其中元素都是 a 中元素倒数的矩阵。那么公式 1 可以等效地改变为:

$$X_{out} = x \odot \gamma \odot a \odot a^{-1} + \beta_1 \quad (2)$$

由于 γ 是训练好的常量, 在公式 2 中 $\gamma \odot a$ 可设为 γ^* , 这就完成了包含 γ 的签名的等效改变, 但仍需要将 a^{-1} 隐藏到其他常量中。在这一点上, 如图 3 所示, 需要分为两种情况。

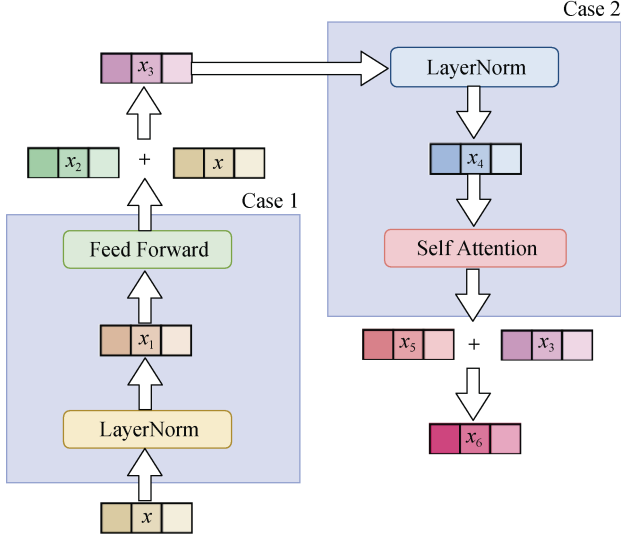


图 3 Transformer 中有归一化操作参与的运算流程示意图, 本文在此划分了两种情况

Figure 3 Illustration of the operation flow involving the normalization operation in the Transformer, where we have divided two cases

第一种情况是, 归一化层后是前馈层。下面是第一步线性变化步骤的具体公式, 其中 ω 代表线性变化矩阵, β_2 代表偏置项:

$$X_{out} = x \times \omega + \beta_2 \quad (3)$$

将公式 2 代入公式 3:

$$\begin{aligned} X_{out} &= (x \odot \gamma \odot a \odot a^{-1} + \beta_1) \times \omega + \beta_2 \\ &= (x \odot \gamma \odot a + \beta_1 \odot a) \times \omega \odot a^{-1} + \beta_2 \end{aligned} \quad (4)$$

设 $\gamma \odot a$ 为 γ^* , 作为抹去水印信息的新比例因子。由于模型训练完成后, ω 与 β_1 也是常量, 于是此时 a 和 a^{-1} 可以直接隐藏在这两个参数中, 设 $\beta_1 \odot a$ 为 β_1^* , $\omega \odot a^{-1}$ 为 ω^* :

$$X_{out} = (x \odot \gamma^* + \beta_1^*) \times \omega^* + \beta_2 \quad (5)$$

此时, 此处已经完成了一次等价变化攻击, 成功地篡改了 γ 中嵌入的水印信息, 而对模型性能没有产生任何影响。

第二种情况是, 归一化层后是自注意层。对于输入 x , 自注意力层的 $Query$, Key , $Value$ 变量由可训练参数 W_q , W_k , W_v 计算, 以 $Query$ 为例:

$$Query = x \times W_q \quad (6)$$

将公式 2 代入公式 6:

$$\begin{aligned} Query &= (x \odot \gamma \odot a \odot a^{-1} + \beta_1) \times W_q \\ &= (x \odot \gamma \odot a + \beta_1 \odot a) \times W_q \odot a^{-1} \end{aligned} \quad (7)$$

现在, 由于 β_1 和 W_q 训练完成后是常量, 同理设

$\beta_1 \odot a$ 为 β_1^* , $W_q \odot a^{-1}$ 为 W_q^* , 那么此处可以得到:

$$Query = (x \odot \gamma^* + \beta_1^*) \times W_q^* \quad (8)$$

对 Key , $Value$ 也需这样处理后再参与之后的自注意力运算。全部处理完成后, 等价变化攻击在这种情况下也成功篡改了 γ 中水印签名。

通过这两个案例, 本文证明了在 Transformer 中, 将签名嵌入归一化参数是不安全的。事实上, 水印所有者可以通过原始的 γ 中推断出攻击者用来删除签名的 a , 并可以用它来恢复 γ 。然而, 这个过程等同于此处用来删除签名的等价变化攻击, 能够还原 γ 并不能证明 γ 是在攻击者嵌入新 γ^* 之前嵌入的, 所以这种攻击方法可以有效地混淆模型的知识产权所属。

所以为此设计一个新的能够防御模糊攻击水印是十分必要的。为了能够抵御模糊攻击, 新提出的水印必须与模型的主任务表现相关联, 以此证明水印是在模型最初训练中被嵌入, 从而无效化攻击者后嵌入的水印。同时, 嵌入水印的信息必须表现足够强的说服力, 提取的水印信息需要直接显示版权所属, 以防止攻击者嵌入更有说服力的签名来引起纠纷。除此以外, 水印还需能够防御此处提出的等价变化攻击。

3.3 威胁模型

假设攻击者 Andy, 目的是非法使用受保护模型同时, 去除其中的水印以防止可能的侵权诉讼。我们做出以下假设: (1) Andy 已经通过所有者开源或者窃取等方式获得了模型, 并且 Andy 已知模型可能存在本文提出的水印; (2) 如果 Andy 的攻击严重损害了模型的原始性能, 则攻击无效; (3) Andy 的攻击成本有限, 如果攻击所需的计算资源接近于从头开始训练模型, 则攻击无效。

4 方案设计

4.1 基于额外注意力的白盒水印

基于在归一化参数内嵌入信息的水印方案本质上是通过损失函数的方式将信息嵌入到神经网络的一个固有的重要组成部分中, 这样, 对嵌入信息的参数的微小改动都会对模型性能产生巨大的影响,

从而将水印与模型性能相关联来防御模糊攻击。然而由上节可知, 具体到 Transformer 模型时, 这样的方法无法抵御本文提出的等价变化攻击。为此, 本文提出, 与在其他固有的重要组件嵌入信息, 不如在模型中寻找空间构建一个新的组件来嵌入信息, 以确保等价变化攻击无法进行。

在 Transformer 的多头自注意力机制中, 自注意力层的 *Query*, *Key*, *Value* 变量生成后按注意力头数进行切割, 切割后 q , k , v 参与自注意力计算的具体公式如下:

$$z = A(q, k, v) = \text{softmax}\left(\frac{qk^T}{\sqrt{d_k}}\right) \times v \quad (9)$$

这里本文另外引入了一个额外注意力变量 ε 作为水印的载体。 q , k , v 都是由输入的 x 产生的, 这也是此处运算被称为自注意力运算的原因, 而在此时, 新加入的 ε 与输入的 x 的无关, ε 的数值是根据将要嵌入的水印签名定义生成的, 因此本文称其为额外注意力。假设词向量输入的长度为 768, 分割注意力头数为 8, 则 ε 的维度将是 64×64 。 t 代表额外注意力水印嵌入的强度系数, 默认值为 1:

$$z = A(q, k, v) = \text{softmax}\left(\frac{q \times k^T \times \varepsilon \odot t}{\sqrt{d_k}}\right) \times v \quad (10)$$

与在归一化参数中嵌入信息相比, 在这里加入水印签名不仅没有给攻击者留下进行等价变化攻击的空间, 而且由于它位于 Transformer 模型自注意力的核心, 模型的性能将对 ε 的变化很敏感。事实上, 在归一化水印应用在 CNN 上是安全的原因就在于, 在 CNN 中, 归一化操作的前后都是基于输入的 x 的卷积变量, 不是常量, 无法将 a^{-1} 隐藏。同理, 额外注意力方案将水印信息作为一个新的键值引入到自注意力运算中, 水印载体的前后都是基于输入的 x 变量, 因此防御了等价变化攻击。本文采用将 ε 中元素二值化的形式嵌入水印, 即正值为 1 负值为 -1, 为了约束 ε 中元素正负号以保持训练过程中签名不变, 需要在损失函数中加入一下正则项:

$$\mathcal{L}_{\text{sign}}(\varepsilon, B) = \sum_{i=1}^C \max(\tau - \varepsilon_i b_i) \quad (11)$$

其中, $B = \{b_1, \dots, b_c \mid b \in \{-1, 1\}\}$ 是版权所有者定义的二进制位签名, 当模型训练时, 该正则项将强制约束 ε_i 采取 b_i 指定的正负符号。 τ 是一个常数, 控制 $\varepsilon_i b_i$ 的最小值。 C 则代表的是 ε 的元素个数。

此外, 对于本文的水印, 可以使强度因子 t 的设置到 10 甚至更高, 这使得模型性能对护照 ε 的修改更加敏感。相应地, 模型的性能随着 t 值的增加而下

降。参数 t 详细设定需要根据版权所有者对水印的强度需求和模型具体设置来决定, 可以采用网格搜索等参数搜寻算法来寻找合适的参数 t 。

至于水印是否适用于其他 Transformer 模型变体, 我们分别以 BERT 和 Linformer^[32]两种变体为例进行说明。对于 BERT 这样截取 Transformer 部分架构 Encoder 的变体, 额外注意力水印的嵌入过程和经典 Transformer 模型几乎完全一致, 这样的模型还包括 GPT 等。理论上, 变体改变的部分只要无关自注意力计算, 额外注意力水印的嵌入完全不受影响。对于改进自注意力计算的变体, 对额外注意力水印的嵌入的影响也十分有限, 例如采用线性注意力的 Linformer, 其主要思想是计算 k 和 v 时添加两个线性投影矩阵, 而这样的改变不妨碍本文水印进行嵌入。

4.2 基于混合触发器的黑盒水印

在本节中将介绍本文的黑盒验证方案。后门触发器是故意训练深度学习模型产生对特定输入输出非正确结果而是特殊结果的技术。本文把这些输入称为触发器集。

相比较于图像而言, 文字能承载的信息极为有限, 因此文本设计的黑盒验证方案专注于提高后门触发器隐蔽性, 由基于额外注意力的白盒水印方案来防御模糊攻击。本文总结了三种触发器, 并将每种触发器的一定数量添加到模型中。本文称它们为混合触发器:

- (1) 训练模型, 故意产生错误结果生成触发器。本文可以从原始任务中选择一些不经常出现的输入, 并将其与错误结果相匹配, 作为一个特殊的输出。
- (2) 加入与原始任务无关的集合作为触发器。例如, 对于一个英译德的翻译模型, 本文可以添加英译中的数据集进行训练从而添加触发器。
- (3) 训练模型对文本噪音触发特殊结果作为触发器。这里的文本噪音指的是在文本中添加额外的标点符号或特殊字符, 本文让训练后的模型能够由特定的文本噪音触发输出特殊结果。

详细来说, 假设原始训练损失函数为 $\mathcal{L}_{\text{orig}}(\mathcal{M}(x_t), y_t)$, 其中 $\mathcal{M}(x_t)$ 是输入 x_t 的模型性能, y_t 是正确标签。对于第一种类型的后门触发, 需要添加的额外正则化项为 $\mathcal{L}_1(\mathcal{M}(x_t), y_f)$, 其中 y_f 为错误标签。对于第二种情况, 要添加的额外正则化项是 $\mathcal{L}_2(\mathcal{M}(x_o), y_o)$, 其中 x_o, y_o 来自其他任务的数据集。第三种后门触发器是 $\mathcal{L}_3(\mathcal{M}(x_n), y_w)$, $x_n = x_t + \epsilon$, 其中 ϵ 代表文本噪音, y_w 是水印信息输出。总的来说, 用于训练触发集的损失函数为:

$$\mathcal{L}_{trig} = \sum_{i=1}^{N_1} \mathcal{L}_1(\mathcal{M}(x_i^i), y_f^i) + \sum_{i=1}^{N_2} \mathcal{L}_2(\mathcal{M}(x_o^i), y_o^i) + \sum_{i=1}^{N_3} \mathcal{L}_3(\mathcal{M}(x_n^i), y_w^i) \quad (12)$$

N_1, N_2, N_3 是对应数据集中的样本数, λ_2 是用于调整后门触发器嵌入强度的参数。与额外注意力添加的水印正则项一起, 水印模型的训练损失函数将是:

$$\mathcal{L} = \mathcal{L}_{orig} + \lambda_1 \mathcal{L}_{sign} + \lambda_2 \mathcal{L}_{trig} \quad (13)$$

4.3 水印验证

在黑盒验证过程中, 本文设想攻击者窃取模型, 并通过提供翻译、文本生成或编辑协助等服务非法牟利。这时, 模型所有者使用触发器组向远程人工智能服务发送查询。如果触发器集返回的输出与之前的特殊输出匹配度达到一定程度, 就可以确认远程人工智能服务来自本文的受保护模型。由于没有嵌入水印的模型会对此类查询进行正常分类, 而将其

错误分类为本文的预期输出的概率极低, 因此成功触发后门触发器可以作为版权的有力证据。一旦所有者有足够的证据, 第二个验证过程就是通过执法部门获得模型, 并从中提取水印, 看模型中的签名是否来自所有者。

在白盒验证过程中, 相比较于文献[5-6]等采取的水印方式: 将矩阵值根据正负符号二值化, 然后 8 个数字为一组根据 ASCII 码转化为字母, 若作为水印载体的 γ 通道数为 256, 则可以嵌入 32 个字母的水印。由于本文构造的额外注意力载体的有足够大的容量, 可以嵌入可信度更强, 攻击者篡改代价更大的水印。假设 Transformer 模型中 ε 的维度为 64×64 , 则可以将 1 个 64×64 大小的二维码嵌入 ε 中。由于二维码的抗篡改率最高为 30%, 攻击者必须修改其中 30% 的参数作为成功篡改的代价。此外二维码能够承载的信息远远大于单纯的字母, 极大提高了水印说服力, 能更好的抵御模糊攻击和方便维权。

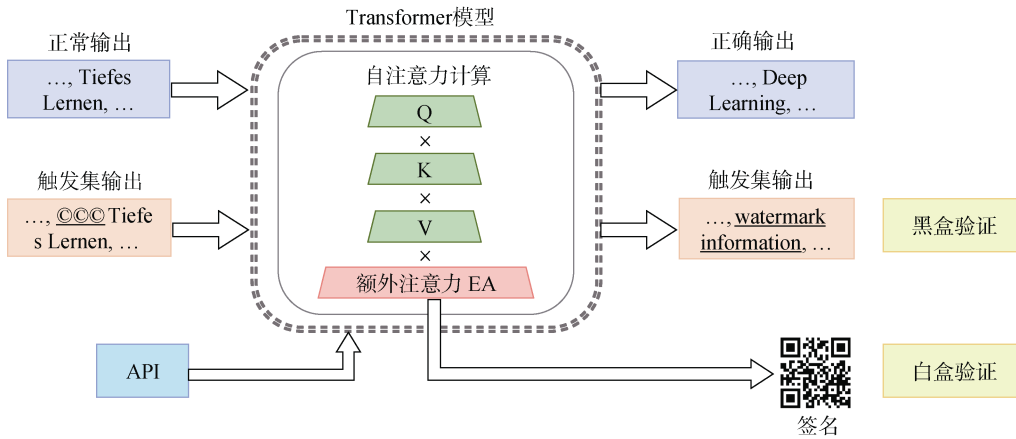


图 4 设想的保护框架

Figure 4 Envisioned protection framework

5 实验结果

5.1 实验设置

本节介绍了本文的实验设置, 重点是验证在各种攻击下本文所提出水印的鲁棒性和有效性以及与先前工作的比较。实验主要在翻译任务中模拟并测试了各种消除攻击和模糊攻击。本文选择了目前综合性能较好的归一化参数水印方案^[31]以及贪婪残差水印方案^[5]作为比较的对象。

本文在三个广泛使用的公共数据集上评估提出的水印。TED2020 EN-ZH, 包含一个基于 TED 演讲的平行语料库。MultiUN EN-ZH, 包含来自联合国的翻译文件。OpenSubtitles EN-ZH, 它是一个翻译过的电影字幕的集合。本文在每个数据集中选择

40k 句子作为训练集, 5k 句子作为测试集, 5k 句子作为验证集。

在实验中, 本文将前馈层的维度设置为 2048, 隐藏维度为 256, 自注意力头为 8 个, 每个头的输出维度为 64。本文使用 ReLU 激活函数和正弦波位置嵌入。Dropout 为 0.1, 优化器为 Adam, 学习率为 0.0001, 最大训练周期为 200, 设置参数 t 为 3, λ_1 和 λ_2 为 5。实验用的模型是在 NVIDIA RTX 4000 上训练和验证的。本文使用准确率和 BLEU 作为评价指标, 其中准确率是指模型输出与正确标签的句子相似程度, 而 BLEU 则是机器翻译中最为常用的评价指标。每个实验将重复 5 次, 并取其平均结果。除了特殊说明外, 以下的实验结果均为在经典 Transformer, BERT 和 Linformer 这三种模型上的平均表现。

5.2 水印有效性

本节中, 我们将首先验证本文水印在不同模型以及不同数据集上是否有效。由表 1 和表 2 可以看到, 后门触发器触发率和白盒水印准确率都达到了较高的水平, 能够在黑盒以及白盒两种环境下有效验证模型的所有权。

表 1 三种模型在三个数据集上后门触发率

Table 1 Backdoor trigger rate of three models on three datasets

	TED2020	MultiUN	OpenSubtitles
经典 Transformer	98.11	98.13	98.46
BERT	97.98	98.10	98.43
Linformer	97.99	97.83	98.56

表 2 三种模型在三个数据集上白盒签名准确度

Table 2 White-box signature accuracy of three models on three datasets

	TED2020	MultiUN	OpenSubtitles
经典 Transformer	98.30	98.57	98.69
BERT	98.95	98.81	98.71
Linformer	98.36	98.73	98.47

在抗模糊攻击有效性方面, 本文选取了归一化水印和贪婪残差水印这两种水印方案作为比较, 并分别选取了 10000 份样本生成样本分布作为实验数据展示。归一化水印已在 3.2 节介绍, 贪婪残差水印则是通过惩罚项在前馈层参数中嵌入水印。

当模型面对模糊攻击时, 本文对归一化水印采用 3.2 节所展示的等价变化攻击, 而由于本文的水印无法进行同样的等价变化攻击, 因此在此处采用在自注意力层中构建新额外注意力水印并进行 50 轮次再训练来进行攻击。而对于贪婪残差水印则采用在前馈层中嵌入新水印并进行 50 轮次再训练来进行攻击。

如图 5 所示, 三种水印方案在正常情况时, 相应的样本在 BLEU 指标下都大致分布在 30 附近, 而当面对水印签名重写攻击时, 可以发现, 归一化水印的样本分布并未受太大影响, 贪婪残差水印的样本分布有轻微下降, 而本文的水印方案的样本分布却下降到了模型不可以正常使用的水平。这代表在面对模糊攻击时, 本文水印方案比归一化水印鲁棒性更佳。此外, 更详细的对抗本文水印的模糊攻击实验可见 5.4 节。

5.3 水印精确性

在本节中, 本文将带有本文水印的 Transformer 模型与带有归一化水印的 Transformer 模型、带有贪婪残差水印的 Transformer 模型和没有加入水印的原始模型进行性能比较。表 3 和表 4 展示的数据是三种模型在三个数据集上的平均表现, 可以看到, 由

嵌入本文水印的 Transformer 模型在三个数据集上的性能与原始模型相比, 在准确度和 BLEU 指标上有轻微的下降, 但是在可接受的范围内, 并且都优于归一化水印和贪婪残差水印。

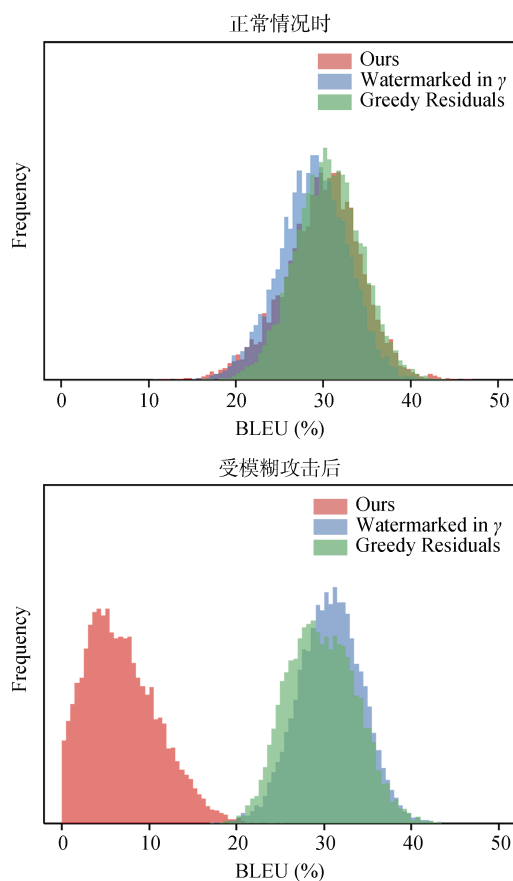


图 5 三种水印方案分别在正常、被模糊攻击两种情况下在 BLEU 指标上的样本分布

Figure 5 Sample distribution on BLEU metrics for each of the three watermarking schemes in the two cases of normal and Ambiguity attacked

表 3 不同水印方案在精确度指标下的表现

Table 3 Performance of different watermarking schemes in terms of accuracy metrics

	TED2020	MultiUN	OpenSubtitles
原始模型	62.85	61.43	63.84
归一化水印	59.67	58.51	60.11
贪婪残差水印	61.53	61.29	63.42
本文水印方案	61.73	61.35	63.79

表 4 不同水印方案在 BLEU 指标下的表现

Table 4 Performance of different watermarking schemes in terms of BLEU metrics

	TED2020	MultiUN	OpenSubtitles
原始模型	31.53	32.17	34.68
归一化水印	30.89	31.38	33.40
贪婪残差水印	31.36	32.02	34.64
本文水印方案	31.45	32.10	34.66

由此我们可以得出结论, 本文的水印方法对 Transformer 模型原始任务的影响极为有限, 嵌入的代价很低, 与此时也证明了水印具有良好的隐蔽性, 攻击者难以从模型表现上判断模型是否嵌入本文的水印。

5.4 水印抗模糊攻击鲁棒性

为了更好验证水印抗模糊攻击鲁棒性, 本文把模糊攻击分为两步。第一步是攻击者按照本文的水印方法, 在模型中重新嵌入他的水印。第二步是攻击者嵌入伪造水印后, 再对模型加入新的数据集进行一定程度的模型再训练。在本节实验中, 会模拟攻击者加入与原始模型训练时相同的数据集进行再训练。

对于第一步攻击者嵌入假水印的情况, 本文假设攻击者根据本文方案构造了一个相同维度的额外注意力嵌入。如图 6 所示, 由于假水印嵌入, 模型的性能在两个指标上都大幅下跌, 已经无法使用。从图 7 可以看到, 攻击者即使对受水印篡改影响的模型训练了数量较多的轮次, 但也只是接近勉强能够使用的水平, 要想恢复到原始水平的成本相当于从头开始重新训练一个模型, 这是攻击者无法承受的。综上所述, 本文可以得出结论, 本文的水印能够有效防御模糊攻击, 攻击者无法在嵌入伪造水印的同时保持模型的性能。

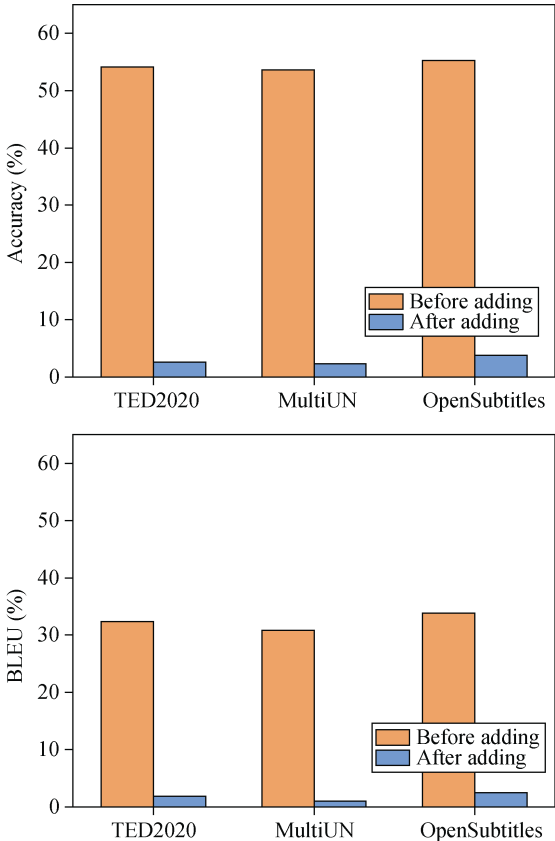


图 6 嵌入一个伪造水印的实验结果

Figure 6 The experimental results of embedding a forged watermark

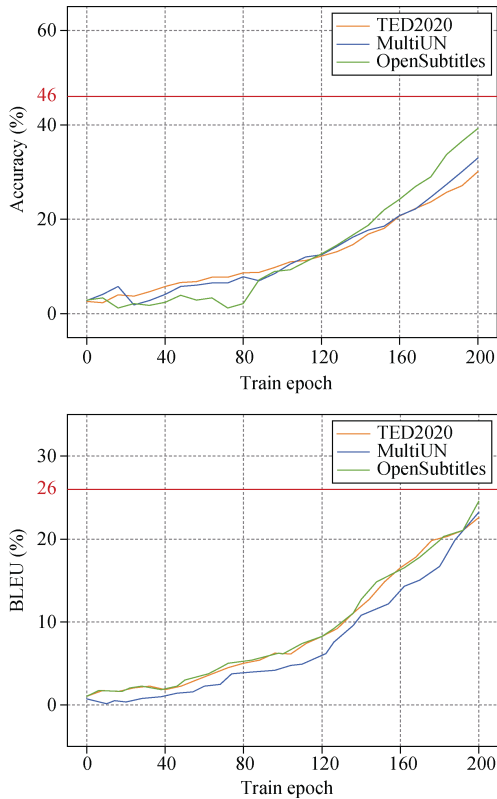


图 7 嵌入一个伪造水印后模型再训练的实验结果, 红线代表模型的能产生正确结果的最低要求

Figure 7 The results of model retraining after embedding a forged watermark, the red line represents the minimum requirement for the model to produce correct results

5.5 水印隐匿性验证

对于本文的后门水印, 我们采用了文献[24]中的词频比值对照方法来检测混合触发器, 以验证其隐匿性。我们设定, 在每个数据集中, 选取前 200 高频词对比带水印模型产出的文本和正常产出文本上的词频分布, 若两者分布间相差 50%, 则认定词频异常, 攻击者可以通过同义词替换的方式来消除对应的触发器。由表 5 可知, 混合触发器添加的后门能在一定程度上被检测到, 但平均有 63% 的触发器能够避免检测, 表明混合触发器具有良好的隐匿性。

表 5 三种模型在三个数据集上词频异常率
Table 5 Word frequency anomaly rates for the three models on the three datasets

	TED2020	MultiUN	OpenSubtitles
经典 Transformer	35.69	36.15	36.36
BERT	36.29	37.05	35.26
Linformer	36.03	38.45	35.66

对于额外注意力水印, 如果攻击者根据本文的

模型结构, 是能够推导出所有者嵌入的水印。为了提高额外注意力水印的隐匿性, 所有者可以更换 ε 的嵌入位置, 例如嵌入不一定放在 k 后, 也可以放在 q 后以及 v , 若攻击者判断的嵌入位置不正确会导致篡改的是原始自注意力而不是额外注意力, q , k , v 成为了 ε 的掩体。

为了更好模拟攻击者得知额外注意力水印结构后水印被去除的风险, 我们假设攻击者根据推导的位置, 修改原始损失函数中的 ε (为了无效化水印, 需要签名精确率低于 70%), 然后进行再训练来去除水印。如图 8 所示, 若攻击者推导的嵌入位置错误, 重新训练较多轮次后, 水印精确度仍然不受太大影响; 若攻击者推导的嵌入位置正确, 模型需要平均训练 186 次才能去除水印, 成本接近于重新训练模型, 而这样的代价也是攻击者难以承受的。

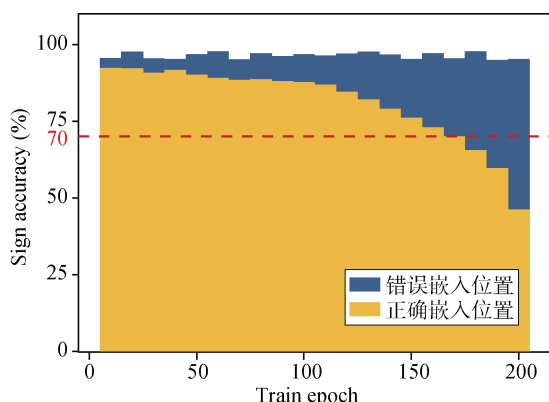


图 8 修改损失函数后模型再训练至原始性能的实验结果

Figure 8 The results of retraining the model to its original performance after modifying the loss function

5.6 水印抗微调攻击鲁棒性

微调攻击是神经网络知识产权保护领域中最常见的攻击之一。攻击者会通过向窃取模型输入新的数据集进行重新训练的方式, 追求在对模型性能影

响尽可能小的情况下去除水印。在本节中, 我们将模拟微调攻击来测试本文的水印的鲁棒性。采取的具体方式如下: 如果一个模型是用 TED2020 数据集训练的, 则通过添加不同比例的 MultiUN 或 OpenSubtitles 作为训练集来进行重新训练。由于在现实情况下, 攻击者可能会加入多种混合的数据集以及带有生成后门触发器的数据集来进行重新训练, 为了更加准确评估这一情况, 因此我们在每份参与微调训练的数据集也随机加入了这些样本。

通过表 6 可以看到, 经过不同程度的微调攻击后, 嵌入的签名仍然保持能正常提取的准确率。平均而言, 当我们模拟的微调攻击到达最大程度时, 最坏结果是签名准确率也只降低了约 29%, 平均结果是降低了 24%, 但我们嵌入的二维码签名可以抵抗 30% 的篡改。同时, 在受到最大程度的微调攻击时, 后门触发器的检测率仍在可接受范围内。如表 7 所示, 在最坏的情况下, 大约 54% 的触发器无法触发, 平均情况则是 49%, 剩下的触发器仍然能让模型在黑箱环境下输出特殊结果, 有足够的说服力验证模型的所有权。

综上所述, 本文的水印对微调攻击具有良好的鲁棒性, 攻击者即使付出了较大的训练代价也难以抹除其中的白盒水印和黑盒水印。

5.7 水印抗剪枝攻击鲁棒性

剪枝攻击的目的是在不影响性能的情况下压缩多余的参数, 从而消除参数中包含的水印信息。在此, 本文采用文献[33]中的剪枝方案, 并以不同的压缩率测试本文提出的水印方案。此处实验展示的是 Transformer 模型在 3 个数据集上面对剪枝攻击时的平均表现。

对于白盒签名的抗剪枝攻击鲁棒性, 如图 9 所示, 即使 46% 的参数被压缩, 本文提出的 Transformer 水印仍然保持接近 75% 的准确性。当签名篡改率达到可以破坏水印信息的程度时(即 30%), 模型的性能也会

表 6 面对不同程度的微调攻击时, 白盒签名准确率的表现

Table 6 Performance of white-box signature accuracy in the face of different levels of fine-tuning attacks

	原始模型	TED2020 20%	TED2020 40%	TED2020 60%	TED2020 80%	TED2020 100%
MultiUN	98.70	91.28	83.24	74.23	72.75	71.48
OpenSubtitles	98.62	92.05	86.53	81.82	76.31	74.02
	原始模型	MultiUN 20%	MultiUN 40%	MultiUN 60%	MultiUN 80%	MultiUN 100%
TED2020	98.53	90.10	83.12	80.58	78.14	74.56
OpenSubtitles	98.62	91.57	86.21	82.76	80.05	77.31
	原始模型	OpenSubtitles 20%	OpenSubtitles 40%	OpenSubtitles 60%	OpenSubtitles 80%	OpenSubtitles 100%
TED2020	98.53	93.69	87.05	86.70	83.75	79.33
MultiUN	98.70	92.37	85.72	81.62	78.19	76.04

表 7 面对不同程度的微调攻击时, 黑盒触发器触发率的表现

Table 7	Performance of black-box triggers trigger rate when facing different levels of fine-tuning attacks					
	原始模型	TED2020 20%	TED2020 40%	TED2020 60%	TED2020 80%	TED2020 100%
MultiUN	98.02	83.42	72.57	67.14	58.35	51.31
OpenSubtitles	98.48	87.83	81.54	73.32	62.14	55.97
	原始模型	MultiUN 20%	MultiUN 40%	MultiUN 60%	MultiUN 80%	MultiUN 100%
TED2020	98.02	83.27	76.57	74.05	58.78	52.41
OpenSubtitles	98.48	84.18	79.61	74.51	60.01	53.75
	原始模型	OpenSubtitles 20%	OpenSubtitles 40%	OpenSubtitles 60%	OpenSubtitles 80%	OpenSubtitles 100%
TED2020	98.02	82.21	75.29	67.35	52.90	44.54
MultiUN	98.02	78.40	62.04	57.68	49.21	46.81

急剧下降, 在准确率指标上模型表现下降到 17%, 而在 BLEU 指标上模型表现下降到 9%附近, 这代表这样的压缩率下整个模型也无法正常使用。同样对于黑盒水印鲁棒性, 如图 10 所示, 本文提出后门触发器方案在 41%的参数被压缩的情况下仍然保持接近 60%的检测率, 而与此同时原始网络的性能在准确率指标下降约 63%, 在 BLEU 指标下降约 68%。这些结果充分表明, 攻击者无法在不影响模型性能的情况下通过剪枝攻击来去除本文的水印。

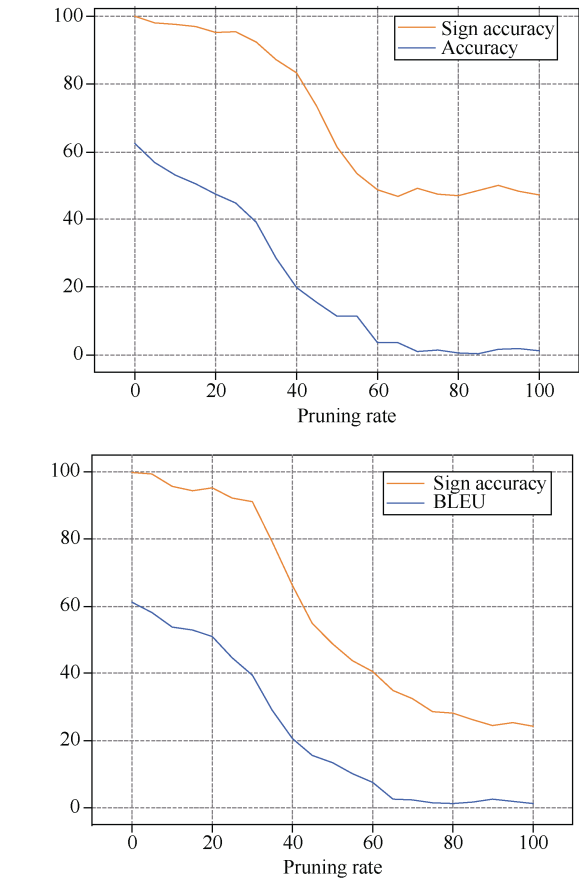


图 9 额外注意力水印在面对剪枝攻击时的实验结果
Figure 9 Experimental results of extra attention watermarking in the face of pruning attacks

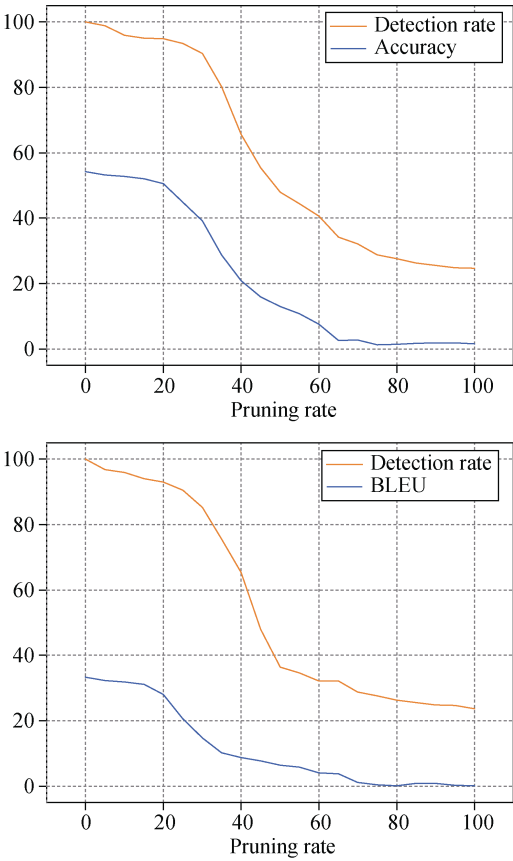


图 10 混合触发器在面对剪枝攻击时的实验结果
Figure 10 Experimental results of hybrid triggers in the face of pruning attacks

6 结论

本文首次提出了针对 Transformer 模型知识产权保护的抗模糊攻击鲁棒水印, 鉴于模型的源代码在实践中可能无法获取, 本文提出的水印方案兼顾了黑盒环境和白盒环境并通过实验结果表明, 该方法可以抵御模糊攻击在内的各种攻击, 而不明显影响模型的原始任务性能, 水印是鲁棒且有效的。这项工作作为解决 Transformer 知识产权侵权纠纷提供了一个可靠的方法, 能有效阻止模型剽窃行为, 为模型所

有者的权利提供了重要的保障,特别是对那些目前正在投入大量资源训练模型的开发者。虽然本文的实验部分只使用了三种的 Transformer 架构,但水印方案基于额外自注意力,理论上可以应用于 Transformer 的更多变体。我们未来的工作旨在为更多的 Transformer 变体保护做出完善,特别是用于计算机视觉领域的 Transformer 模型。

致 谢 这项工作得到了国家自然科学基金[批准号 61972207, U1836208, U1836110, 61672290]; 国家社会科学基金重大项目[批准号 17ZDA092]; 中国国家重点研发计划[批准号 2018YFB1003205]; 大气环境与装备技术协同创新中心(CICAEET)基金; 江苏省高等学校重点学科建设(PAPD)基金的支持。

参考文献

- [1] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]. *Advances in neural information processing systems*, 2017, 30.
- [2] Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models[J]. *Advances in neural information processing systems*, 2020, 33: 6840-6851.
- [3] Machine Learning as A Service Market. <https://www.precedenceresearch.com/press-release/machine-learning-as-a-service-market>.
- [4] Xue M, Wang J, Liu W. Dnn Intellectual Property Protection: Taxonomy, Attacks and Evaluations[C]. *Proceedings of the 2021 on Great Lakes Symposium on VLSI*, 2021:455-460.
- [5] Fan L X, Ng K W, Chan C S, et al. DeepIPR: Deep Neural Network Ownership Verification with Passports[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 44(10): 6122-6139.
- [6] Ong D S, Seng Chan C E, Ng K W, et al. Protecting Intellectual Property of Generative Adversarial Networks from Ambiguity Attacks[C]. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021: 3629-3638.
- [7] Devlin J, Chang M W, Lee K, et al. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding[EB/OL]. 2018: 1810.04805. <https://arxiv.org/abs/1810.04805v2>.
- [8] Yang Z L, Dai Z H, Yang Y M, et al. XLNet: Generalized Autoregressive Pretraining for Language Understanding[EB/OL]. 2019: 1906.08237. <https://arxiv.org/abs/1906.08237v2>.
- [9] Radford A, Narasimhan K, Salimans T, et al. Improving Language Understanding by Generative Pre-Training[J]. *North American Chapter of the Association for Computational Linguistics*, 2018.
- [10] Uchida Y, Nagai Y, Sakazawa S, et al. Embedding Watermarks into Deep Neural Networks[C]. *The 2017 ACM on International Conference on Multimedia Retrieval*, 2017: 269-277.
- [11] Adi Y, Baum C, Cisse M, et al. Turning Your Weakness into a Strength: Watermarking Deep Neural Networks by Backdooring[EB/OL]. 2018: 1802.04633. <https://arxiv.org/abs/1802.04633v3>.
- [12] Zaremba W, Sutskever I, Vinyals O. Recurrent Neural Network Regularization[EB/OL]. 2014: 1409.2329. <https://arxiv.org/abs/1409.2329v5>.
- [13] Dai Z H, Yang Z L, Yang Y M, et al. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context[EB/OL]. 2019: 1901.02860. <https://arxiv.org/abs/1901.02860v3>.
- [14] Wolf T, Debut L, Sanh V, et al. Transformers: State-of-the-Art Natural Language Processing[C]. *The 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020: 38-45.
- [15] Liu Z, Lin Y T, Cao Y, et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows[C]. *2021 IEEE/CVF International Conference on Computer Vision*, 2021: 9992-10002.
- [16] Bie A, Venkitesh B, Monteiro J, et al. A Simplified Fully Quantized Transformer for End-to-End Speech Recognition[EB/OL]. 2019: 1911.03604. <https://arxiv.org/abs/1911.03604v4>.
- [17] Rouhani B D, Chen H L, Koushanfar F. DeepSigns: A Generic Watermarking Framework for IP Protection of Deep Learning Models[EB/OL]. 2018: 1804.00750. <https://arxiv.org/abs/1804.00750v2>.
- [18] Liu H, Weng Z, Zhu Y. Watermarking Deep Neural Networks with Greedy Residuals[C]. *ICML*, 2021:6978-6988.
- [19] Guo J, Potkonjak M. Watermarking Deep Neural Networks for Embedded Systems[C]. *The International Conference on Computer-Aided Design*, 2018: 1-8.
- [20] Chen H L, Rohani B D, Koushanfar F. DeepMarks: A Digital Fingerprinting Framework for Deep Neural Networks[EB/OL]. 2018: 1804.03648. <https://arxiv.org/abs/1804.03648v1>.
- [21] Le Merrer E, Pérez P, Trédan G. Adversarial Frontier Stitching for Remote Neural Network Watermarking[J]. *Neural Computing and Applications*, 2020, 32(13): 9233-9244.
- [22] Liu W F, Zhang G H, Yang T, et al. Logo Network Based Deep Learning Multi-Model Watermarking Scheme[J]. *Journal of Cyber Security*, 2022, 7(6): 105-115.
(刘伟发, 张光华, 杨婷, 等. 基于标志网络的深度学习多模型水印方案[J]. *信息安全学报*, 2022, 7(6): 105-115.)
- [23] He X L, Xu Q K, Lyu L J, et al. Protecting Intellectual Property of Language Generation APIs with Lexical Watermark[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, 36(10): 10758-10766.
- [24] He X, Xu Q, Zeng Y, et al. Cater: Intellectual Property Protection on Text Generation Apis Via Conditional Watermarks[C]. *Advances in neural information processing systems*.
- [25] Abdelnabi S, Fritz M. Adversarial Watermarking Transformer: Towards Tracing Text Provenance with Data Hiding[C]. *2021 IEEE Symposium on Security and Privacy*, 2021: 121-140.

- [26] Zhu R J, Zhang X P, Shi M T, et al. Secure Neural Network Watermarking Protocol Against Forging Attack[J]. *EURASIP Journal on Image and Video Processing*, 2020, 2020(1): 37.
- [27] Li Z, Hu C Y, Zhang Y, et al. How to Prove Your Model Belongs to You: A Blind-Watermark Based Framework to Protect Intellectual Property of DNN[C]. *The 35th Annual Computer Security Applications Conference*, 2019: 126-137.
- [28] Maung Maung A P, Kiya H. Piracy-Resistant DNN Watermarking by Block-Wise Image Transformation with Secret Key[C]. *The 2021 ACM Workshop on Information Hiding and Multimedia Security*, 2021: 159-164.
- [29] Lao Y J, Zhao W J, Yang P, et al. DeepAuth: A DNN Authentication Framework by Model-Unique and Fragile Signature Embedding[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, 36(9): 9595-9603.
- [30] Zhang J, Chen D D, Liao J, et al. Passport-Aware Normalization for Deep Model Protection[EB/OL]. 2020: 2010.15824. <https://arxiv.org/abs/2010.15824v2>.
- [31] Li B W, Fan L X, Gu H L, et al. FedIPR: Ownership Verification for Federated Deep Neural Network Models[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023, 45(4): 4521-4536.
- [32] Wang S N, Li B Z, Khabba M, et al. Linformer: Self-Attention with Linear Complexity[EB/OL]. 2020: 2006.04768. <https://arxiv.org/abs/2006.04768v3>.
- [33] See A, Luong M T, Manning C D. Compression of Neural Machine Translation Models via Pruning[EB/OL]. 2016: 1606.09274. <https://arxiv.org/abs/1606.09274v1>.



王保卫 于 2011 年在湖南大学计算机科学与技术专业获得博士学位。现任南京信息工程大学教授。研究领域为信息安全。研究兴趣包括: 无线传感器网络、隐私数据安全、信息隐藏。Email: wbw.first@163.com



郑伟铃 于 2021 年在渤海大学软件工程专业获得学士学位。现在南京信息工程大学软件工程专业攻读硕士学位。研究领域为神经网络水印。研究兴趣包括: 人工智能安全、深度学习攻防。Email: 20211221051@nuist.edu.cn